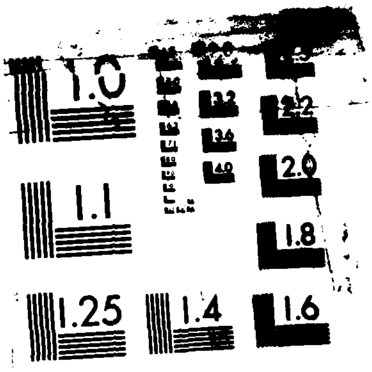
AD-A175 439 CONTROL ELECTRONICS FOR AN ION MASS FILTER IN THE LOWER IONOSPHERE PAYLOA..(U) NORTHEASTERN UNIV BOSTON MASS ELECTRONICS RESEARCH LAB R SUKYS ET AL. 30 SEP 85 1/1

UNCLASSIFIED AFGL-TR-86-0170 F19628-81-C-0162 F/G 14/2 NL

END
DATE
FILMED
2-87



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

AD-A175 439

AFGL-TR-86-0170

CONTROL ELECTRONICS FOR AN ION MASS FILTER
IN THE
LOWER IONOSPHERE PAYLOAD DEVELOPMENT PROGRAM

by

Raimundas Sukys
J. Spencer Rochefort

NORTHEASTERN UNIVERSITY
360 Huntington Avenue
Boston, Massachusetts 02115

DTIC
ELECTE
DEC 30 1986
S D
D

30 September 1985

FINAL REPORT
15 September 1981 - 30 September 1985

Approved for public release: distribution unlimited.

DTIC FILE COPY

Prepared for

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

86 12 29 016

"This technical report has been reviewed and is approved for publication."

Alan Bailey
(Signature)
Alan Bailey
Contract Manager

John E Rasmussen
(Signature)
John Rasmussen
Branch Chief

FOR THE COMMANDER

(Signature)
Robert Skrivanek
Division Director

This report has been reviewed by the ESD Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requestors may obtain additional copies from the Defense Technical Information Center. All others should apply to the National Technical Information Service.

If your address has changed, or if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify AFGL/DAA, Hanscom AFB, MA 01731. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractural obligations or notices on a specific document requires that it be returned.

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADA175439

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFGL-TR-86-0170

6a. NAME OF PERFORMING ORGANIZATION
Northeastern University
Electronics Research Lab.

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Air Force Geophysics Laboratory

6c. ADDRESS (City, State and ZIP Code)
360 Huntington Avenue
Boston, MA 02115

7b. ADDRESS (City, State and ZIP Code)
Hanscom Air Force Base
Bedford, MA 01731

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Air Force Geophysics Lab.

8b. OFFICE SYMBOL (If applicable)
LI

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F19628-81-C-0162

8c. ADDRESS (City, State and ZIP Code)
Hanscom Air Force Base
Bedford, MA 01731

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO
61102F	2310	G3	AW

11. TITLE (Include Security Classification) Control Electronics for an Ion Mass Filter in the Lower Ionosphere Payload Development Program

12. PERSONAL AUTHOR(S)
R. Sukys, J.S. Rochefort

13a. TYPE OF REPORT
Final Report

13b. TIME COVERED
FROM 9/15/81 TO 9/30/85

14. DATE OF REPORT (Yr., Mo., Day)
1985 September 30

15. PAGE COUNT
90

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB GR

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Ion Mass Spectrometer, Control Circuits.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report describes a control and data transmission system for a switchable negative/positive ion mass filter. The system was developed for small diameter rocket applications. A Superarcus rocket could place the mass spectrometer system in the 40 to 60 km range for measurements.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Alan D. Bailey

22b. TELEPHONE NUMBER (Include Area Code)
617-377-2525

22c. OFFICE SYMBOL
AFGL/LID

EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

Accesion For
NTIS CRA&I ☒
DTIC TAB □
Unannounced □
Justification ....
By ....
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1 | LMD 25

QUALITY INSPECTED 4

TABLE OF ILLUSTRATIONS

INTRODUCTION

The contract F19628-81-C-0162 was written for the design, fabrication, modification and testing of prototype research instrument systems for ongoing AFGL measurement programs. Field engineering and technical support during flight systems checks at various field sites and test ranges was also required.

During the life of the contract the emphasis was placed on the design and development of specialized control electronics for ion mass spectrometers. Some work was also done on thermosonde/radiosonde systems for the measurement of the optical atmospheric turbulance. Most of the electronic systems were developed to control airborne instruments. Although a few subsystems were fabricated to control instruments flown on satellites and the shuttle, balloon and sounding rocket applications predominated.

A typical control system for a mass filter included a microprocessor or a microcontroller to manage pre-programmed commands, control parameters and data. Digital to analog interface circuits converted the control parameters into the basic analog signals necessary for the operation of a quadrupole ion mass filter. From these base signals the bias voltages and the quadrupole excitation was generated. For that purpose various amplifier configurations and high voltage supplies were employed. An oscillator whose amplitude could be varied with a great degree of precision over a wide range provided the ac excitation component for the quadrupole mass filter. The spectral data collected by electron multiplier devices, either in a current or a pulse mode, were conditioned by logarithmic current-to-voltage converters or pulse counters respectively. In either case, the data was converted into a PCM bit stream for transmission through a telemetry link. Various monitor and data identification signals were included to facilitate data reduction and interpretation. The control system was powered by a multiple output power converter tailored to the needs of the system.

Variations of the basic approach accommodated special requirements. Electron beam ionization generators and control circuits were added to the instruments intended to measure neutral constituents. Instruments capable of switching between positive and negative ion measurements during a flight were built. Circuits to control the potential difference between the vehicle and the instrument were provided when needed. The PCM data subsystems were omitted from a few of the mass spectrometers.

Ground support equipment was also provided. The support equipment consisted mainly of control consoles for laboratory tests and launch operations. Included among the ground support equipment were units designed to interface with a control and data processing computer used to direct the operation of a balloon-borne instrument system during a flight. These units provided a partial real time data processing that reduced the burden placed on the computer freeing it for a more efficient incoming data analysis necessary for the interaction between the ground based scientist and the airborne instrument. A command interface between the computer and the transmitting equipment of the ground station was also provided by the units.

Some of the development work has been described in Scientific Reports 1, 2 and 3 issued under this contract and listed in the publications section. Other tasks and services rendered were reported only in the Quarterly Status reports or were communicated to the contract monitor as drawings, descriptions, specifications and operating instructions.

This final report describes a control system for a switchable ion mass filter designed as a part of the Lower Ionosphere Payload Development (LIPD) project. The system provided the necessary control functions and bias voltages for a miniature cryogenically pumped ion mass filter intended to make measurements in the 40 to 60 kilometer altitude range. The control circuits included a complete PCM data system. The development has been carried through an operational breadboard stage ready to be tested with the mass filter.

I. LIPD OVERVIEW

The objective of the Lower Ionosphere Payload Development (LIPD) project was to design a lightweight cryogenically pumped Ion Mass Spectrometer for the exploration of the ionosphere at the altitude between 40 and 70 kilometers. The instrument was to be carried on a 11.4cm diameter Superarcus rocket fired from a portable launcher. The Ion Mass Spectrometer, a nosecone ejection mechanism, a battery and the telemetry was to be packaged into a 100cm long payload including a 53cm ogival nosecone and weighing less than 9 kilograms.

The development of the control unit for the positive/negative ion quadrupole mass filter capable of detecting ions up to 150 atomic mass units was assigned to this contract. Telemetry and the mechanical design including payload packaging and integration were the responsibility of the contracts F19628-83-C-0037 and F19628-81-C-0029 respectively. To conserve weight and space, the mass filter control unit also provided the timing signals for the nosecone ejection and formatted the mass filter data and the payload monitor signals into a PCM data stream ready to modulate an FM transmitter. A 200 milliwatt S-band FM transmitter and a stripline band antenna satisfied the TLM requirements. A single lithium battery pack was chosen to provide power to the whole payload. The mass filter, cryogenic pump and the ion/electron detection devices were the responsibility of AFGL.

The design of the mass filter control unit was based on an eight-bit microcontroller with a built in EEPROM. The operating system program was stored in the EEPROM. Flight and/or the test parameters to control the filter were stored in an EEPROM accessible from the outside through the communications port of the microcontroller. A block of 16 eight-bit instructions defined up to three atomic mass units through which the mass filter could be stepped while maintaining the same offsets, biases and the ratio between the ac and the dc components of the quadrupole excitation signal. A total of 126 such blocks were available for the definition of a flight program that could be repeated indefinitely.

Two twelve-bit DAC's were employed to generate the quadrupole excitation control signal and to set the ratio between the ac and the dc components of the excitation. A domain of one atomic mass unit was defined by twenty-seven levels of the control signal. Offset control for the excitation signals was provided by three 8-bit DAC's receiving their inputs from the mass filter control program. Two of the signals were primarily intended to eliminate the offset voltages of the power amplifiers producing the two dc components of the quadrupole excitation. The third signal controlled the offset of the ac component. Four bipolar digitally controlled bias signals generated by 8-bit DAC's and two fixed high voltage supplies to bias the ion detectors completed the list of signals required by the mass filter.

The mass spectrometer dwelled 10ms at each selected atomic mass unit. The data in a form of a pulse count was collected during the last 9ms of the dwell time. The first millisecond was allotted for the stabilization of the quadrupole excitation after the selection. A sixteen-bit counter was used. The data collection process and the PCM telemetry were synchronized. The ion data collected during a PCM frame was transmitted together with the support data during the following frame. Each frame consisted of 20 eight-bit words, MSB first. The word assignments within a frame are tabulated in Appendix A. Data, atomic mass unit identification and the ratio information were transmitted as two consecutive 8-bit words, the most significant byte first. The remaining words carried other support and monitor data. Analog monitor signals were converted into the digital form by an 8-channel data aquisition system. The 16kbps PCM data stream was converted into the Bi-phase Level form for transmission through the FM link.

The ac excitation for the quadrupole was generated by an oscillator whose frequency was determined primarily by a resonant circuit consisting of the secondary winding of the output transformer and the capacitance of the quadrupole. The generator was able to produce an output signal that ranged from 2.5 volts to 450 volts peak at 2.3MHz into a 94pF capacitive load. This amplitude range exceeded the quadrupole requirements to detect ions between 10 and 150 atomic mass units.

An oscillator-driven, non-saturating, dc-to-dc converter provided ±110, ±15, and +5 volt outputs to meet circuit requirements. The converter operated at a nominal 25kHz frequency and required between 0.6 and 1 A at 28 volts. The two current limits occurred when the filter was set to process ions at the two extreme atomic mass units of 10 and 150 respectively. Battery voltage to the converter was pre-regulated at +20 volts.

The instrument could be armed in a preparation area before being transported to the launcher. Applying an arming pulse through a small connector in the skin of the payload latched a relay that connected the battery to the pullaway circuits only. Power to the control circuits and the transmitter remained blocked as long as an external connection carrying approximately 1mA of current from the payload battery to the pullaway circuits was in place. Upon launch the external connection was broken. At that time power became available to the transmitter and to the control circuits. The ac exciter and the HV supplies were activated after a pre-programmed time interval during the flight when the nosecone of the vehicle had been ejected and the mass spectrometer had been exposed to the atmosphere.

II. CIRCUITS

In this chapter a brief description of the circuits in the mass filter control unit is presented. The descriptions include, where appropriate, the function of an individual component and its relationship to other components during the execution of a control task. Whenever possible, a block of circuits contributing to the execution of a given control function or functions are presented together in a single circuit diagram.

A. Digital Circuits

The circuits generating and/or responding to digital commands and intended for packaging as a functional unit are shown in Figure 1. The design of the control unit was based on an INTEL 8751 (U1) microcontroller (uC). The micro-controller operated at 6.144MHz and contained the operating system program in its internal EPROM. The mass filter control program and the timing information was stored in the EEPROM (U2). Multiplexed bus structure was used to address and to transfer data to and from the other integrated circuits. The microcontroller ports zero and two were used for that purpose.

The EEPROM was the only component connected to the bus that required an external address latch. U4 latched the lower byte of the address for the PROM.

The serial I/O port of the microcontroller was utilized in a *full-duplex configuration* to communicate with external devices in an asynchronious mode. Through this port the EEPROM could be programmed and the other devices connected to the bus could be accessed. The remainder of the I/O pins were used to generate individual discrete commands to control other circuits in the payload or to provide the chip select and control functions for other integrated components on the bus. To augment the available microcontroller I/O pins for the chip control functions a 3 to 8 line decoder (U3) was used.

A watchdog circuit was employed to guard against a program crash. The circuit consisted of a counter (1/2 of U13) driven by a 16kHz input and the NAND gates U15X. A pulse generated at Y7 of U3 under the software control of the microcontroller cleared the counter (pin 12 of U13) every 500us. The pulse propagated through U15C and the RC network. Failure to clear the counter produced a positive enabling pulse at pin 8 of U13 and 62.5us later a pulse of the same duration at 10U13. That latter pulse passed through the coincidence gates to the reset pin of the microcontroller. Since the most likely period of time for a noise induced program crash could be anticipated to be during the lift-off, the watchdog circuit could restore the system to a proper operation with a minimum loss of the data window. The reset at power-on was generated by the RC circuit at pin 5 of U15B.

The analog signals to control the quadrupole excitation were generated by 12-bit A/D converters. The converter U16 generated the ac excitation control signal while the U17 established the ratio between the ac and the dc components of the excitation. Each atomic mass unit domain was resolved into 27 levels differing one from the other by 1 significant bit. The converter output covered a nominal range from zero to +10 volts. The -10V reference for the converter was derived from the monolytic source U21.

To generate the four bias signals a quad 8 bit A/D converter U19 was used. Same type of a converter (U18) generated three dc signals to compensate for excitation amplifier offset voltages or to introduce, if needed, some offsets into the excitation signals. To eliminate possible loading problems while operating over a wide range of temperatures, separate reference sources (U22 and U23) were provided for each converter.

The ion spectral data appearing in the form of pulses was accumulated during a 9ms period for each atomic mass unit in the 16-bit binary counter U8 and U9. The counters had the tri-state output capability and, therefore, could be connected directly to the bus. The count was transfered into the PCM data stream every 10ms and, at that time, the counters were also cleared. The various analog monitor signals were converted into an 8-bit digital data by the 8 channel data acquisition component U20. The converter received its 256kHz clock from a crystal oscillator U12.

The 16kHz PCM clock was also derived from the same crystal oscillator through a 4 bit binary scaler (1/2 U13) and the U15D gate. The formatting of the PCM frame was under firmware control. A frame consisted of twenty 8-bit words. The words were loaded into the parallel-to-serial shift register U6 by the microcontroller. The microcontroller was interrupted to load a new word every 500us by a timing signal generated at pin 9 of U13. The same interrupt was also used to control the scan rate of the mass filter. The timing for the nosecone ejection and for the activation of the HV supplies and the ac exciter was based on the accumulated count of the interrupts.

The nosecone ejection commands were transmitted through U5. Four descrete commands were provided. Each line was capable to sink 200mA at 28 volts and was intended to drive a relay. To insure that all lines were in the high impedance state during the power-on interval, the same reset signal used to initialize the microcontroller was employed to clear the relay driver. Only after the reset pulse to the microcontroller had been removed, the clear signal was allowed to decay to zero. The transient suppressor line shown in the figure was connected to the relay power source.

B. The Amplifiers

The circuits used to condition and to amplify the dc signals generated by the digital to analog converters are shown in Figures 2 and 3. The exciter control signals were processed by the circuits of Figure 2 while the bias signals were converted to the required polarity and then amplified to the desired levels by the circuits of Figure 3.

The ac exciter control signal was buffered by the unity gain inverting amplifier A_{31} before being passed on to the ac excitation generating circuits. The signal from the multiplying DAC, that controlled the ratio between the ac and the dc components of the quadrupole excitation, was processed by the amplifier circuits A_{21}. A_{22}, A_1 and A_2. These circuits produced the positive and the negative dc components of the excitation. The two dc signals were very closely matched in magnitude. A common quadrupole bias Q_B was also added to the dc signals through the high voltage amplifiers A_1 and A_2. The offset voltages of the amplifiers could be digitally nulled. The bipolar offset control signals were introduced at the inverting inputs of the amplifiers A_{21}, A_{22} and A_{31}. The first two signals were primarily intended to cancel the dc offsets of the output amplifiers A_1 and A_2. The third signal could be used to manipulate the dc offset requirements of the ac exciter circuits.

One of the four similar bias voltage amplifiers is shown in Figure 3. The unipolar signal generated by an 8 bit DAC was offset and amplified to produce a bipolar signal between -30 and +50 volts with proper choice of R25. (Using 100K as illustrated produces an output bias range of ±50V when the DAC output ranges from 0 to 10V.) MOSFET's were used to boost the operational amplifier outputs to the desired levels. The common supply voltages to all four bias amplifiers were derived from the ±110 volts required by the dc excitation amplifiers.

C. The AC Exciter

The circuits generating the ac component of the quadrupole excitation signal are shown in Figure 4. The opposite phase signals for the two sets of the quadrupole electrodes were obtained from the secondary windings of the oscillator transformer. The free running oscillator design frequency of 2.3MHz was primarily set by the resonant circuit consisting of the output inductance of the windings and a capacitive load. The major contributor to the load capacitance was the quadrupole itself. Additional loading was introduced by the capacitive divider (C5, C7) and C_{TRIM} used to balance the output amplitude at the two windings. The signal to control the amplitude of the oscillator was obtained from the capacitive divider. It was clamped by the circuit of C4, CR7 and CR8. The diode CR8 provided some offset and temperature compensation. The clamped signal was filtered, inverted, attenuated and summed at pin 2 of Al with the exciter control signal. The output of the amplifier provided the drive for Q_1 which in turn controlled the series pass transistor Q_2. This power transistor supplied the collector voltage for the two oscillator drivers Q_3 and Q_4. The dc base drive was also derived from the collector voltage, while the ac feedback signal to the base was obtained through the capicitors C9 and C10 from the feedback windings of the transformer. The transformer was wound on a phenolic toroid 2.4 cm high with the outside diameter of 5cm and an inside diameter of 3.8cm. Amplitude control of the oscillator output could be maintained from a minimum of 2.5 to a maximum of 450 peak volts at a power supply voltage of 22 volts. The current requirements varied between 100 to 500 mA at the two output extremes.

The power to the oscillator could be cut-off by pulling the gate of Q_1 to a ground potential. This circuit was utilized by the digital control subsystem during the initial stages of flight. The oscillator was turned on after the nosecone was ejected. In addition, two protective circuits were introduced into the exciter to interrupt power to the oscillator to prevent damage when a danger to the driver transistors was sensed. One of the protection circuits A3A monitored the oscillator current. When the current exceeded 1A the power was periodically interrupted until the current was reduced. This protection was primarily intended to avoid long periods of a high power dissipation in the transistors when the circuit was accidentally prevented from oscillation. The other circuit (A2A) was tripped by a temperature sensor CR1 when the oscillator base plate temperature exceeded approximately 80°C. The oscillator was activated again when the temperature dropped below 50°C. Amplifier circuits A2B and A3B provided temperature and ac excitation amplitude monitor signals.

D. HV Bias Circuits

The high voltage circuits to bias the Channel Electron Multipliers (CEM) are shown in Figure 5. The two CEM devices, one to measure the positive ions, the other for the negative ion data, were biased by separate HV supplies. The supplies whose outputs were proportional to the input voltages were operated at their maximum output of 3,000 volts. The required input power at 12 volts was derived from the preregulated power supply voltage of 20 volts by the operational amplifier A6X and the two MOSFET's Q_{17} and Q_{18}. The power to the HV supplies could be interrupted by the same circuit (Q_{19}) which controlled the power flow to the ac exciter. Therefore, the HV supplies and the ac exciter were always activated at the same time. Power to the selected supply was switched through a relay which was under the digital circuit control.

The outputs of the HV supplies were connected to the CEM's through two 1M resistors and a capacitor providing some additional filtering of the output ripple. The status of the two outputs were monitored through a 100M resistors terminated by diodes for safety and circuit protection.

A single charge amplifier A2, AMMP-TECH A-101, mounted on the standard PC-11 test board was used to amplify the incoming spectral data. The same relay, which activated the selected HV power supply also switched the amplifier to the appropriate CEM device. Separate ac neutralization circuits were used for the data originating at the two CEM's. The neutralizing signal was derived from the two ac excitation components of the quadrupole. The potentiometer and the centertapped variable capacitor provided the amplitude and the phase control for the neutralizing signals to cancel the interferring ac signal appearing at the input of the amplifier.

E. Support Circuits

Figure 6 is a collection of the various monitor, communications interface and power control circuits.

The arming and power control circuit is shown in the upper left corner of the drawing. The latching relay connected the flight battery to the series pass transistor Q_2 which blocked the power to the rest of the control circuits as long as R_{27} and R_{28} were connected together. When in this configuration, the current drain from the battery was a nominal 2mA. The transistor Q_1 was saturated, Q_3 and Q_2 were cut-off. When the connection between the two resistors was broken Q_1 became cut-off and the pass transistor Q_2 supplied power to the control circuits. Thus, the payload could be armed before the installation into the launcher provided a short between R_{27} and R_{28} was maintained. Closure of the relay could be verified by a voltage measurement. The active OFF circuit was chosen to insure that upon launch the broken safety connection between the two resistors could short to the vehicle without upsetting the operation of the control electronics.

The group of circuits in the lower right part of the drawing are the communications interface circuits. They include the circuits (U1, U2A) to convert the NRZ PCM data into a bi-phase signal suitable to modulate an FM transmitter. The deviation of the transmitter could be adjusted by selecting the resistor R. A monitor output to observe and to use the PCM data stream in the laboratory was also provided (U2B)

The interface circuits to control the operation of the mass spectrometer in the laboratory environment are shown in the lower part of that section. The U2C and U2D circuits were used to interface the microcontroller communications ports with a laboratory control unit. The circuit associated with Q_5 was used to indicate to the microcontroller whether a laboratory test or a flight program was being run. The rest of the circuits shown in the Figure were the various monitors. A_{11} and A_{12} converted the HV monitor currents into the voltage signals suitable for the A to D converter. The absolute value circuit A_{23}, A_{24} and A_{14} monitored the combined bias voltages. Both dc components of the quadrupole excitation signal were monitored as a combined signal by A_{21} and A_{22}. The common bias component Q_B was subtracted in the monitor circuit from the ±DC voltages.The other two circuits (MON 5 and MON 6) were used to monitor the battery voltage and the ±15 volt supply respectively.

F. The Power Supply

The power supply is shown in Figure 7. It was based on a non-saturating squarewave driven transformer design. The pot core transformer was driven by the power MOSFET's Q_1 and Q_2 at approximately 25kHz. The low impedence gate drivers Q_5 to Q_8 received their symetrical base signals from the FF U2 which was clocked by a 50kHz signal generated by U1.

The battery power to the converter was pre-regulated at 20 volts by VR1. VR2 provided the required +15 volts to the circuits directly from the pre-regulated power. All other voltages were derived from the transformer outputs and regulated by the circuits shown.

III. FIRMWARE

The operating system was stored in the EPROM of the microcontroller. Exclusive of the initialization process, the firmware provided three distinct modes of operation. The first mode controlled the payload in the beginning of the flight. At that time the primary task of the microcontroller was to provide the timing for the ejection of the nosecone and the activation of the HV and ac exciter circuits. When that task was completed, the microcontroller entered the data gathering mode. In that mode it provided control signals for the mass filter and formatted the PCM data. The third mode, intended for laboratory use, was command oriented. An external control source could access and modify the existing mass filter control firmware stored in an EEPROM. Also, various other control circuits could be directly accessed. Data read-out could be requested and the operation could be transferred into the data gathering mode. Only the timing functions to eject the nosecone and to automatically activate the HV and the exciter circuits were not accessable for safety reasons.

All communications to the microcontroller from the external source were initiated with a command code. The command was followed by either an address code or a data code or both. The instructions were transmitted at 1200 bits per second using an asynchronious mode. A start bit and one stop bit with no parity were used.

The command codes were 8-bit binary numbers with a ONE in the MSB position. The MSB was used to differentiate between a command and the address or data codes. The addresses and data were transmitted as 8-bit ASCII characters representing the hexadecimal numbers 0 through F. Each character thus defined four binary bits of an address or a data word in the same order of significance as received.

All transmissions to the microcontroller were immediately echoed back for verification. The end of transmission code initiated the execution of the just received command. When a command requested data to be sent back to the external control source, that data was transmitted in the binary code only. All communications from the microcontroller, except for the echo of an "ESCAPE" were followed by the end of transmission code. The command codes and the accessible memory locations are listed in Appendix B.

Upon launch, when the control unit became active, the microcontroller proceeded through an initialization process which included activity to prevent a premature ejection of the nosecone. The externally introduced FLIGHT OR TEST flag was checked. When in the flight mode the microcontroller established a counter and loaded it with the first timing byte stored in the EEPROM. The interrupts from the PCM circuits served as clock pulses to decrement the counter. When the contents of the counter were reduced to zero the second byte was loaded. When the count once again reached zero, a command was generated to eject the nosecone and to remove the seal from the orifice of the mass spectrometer. The primary command was followed by a backup command a short interval later. After an additional delay the HV supplies and the quadrupole exciter circuits were activated.

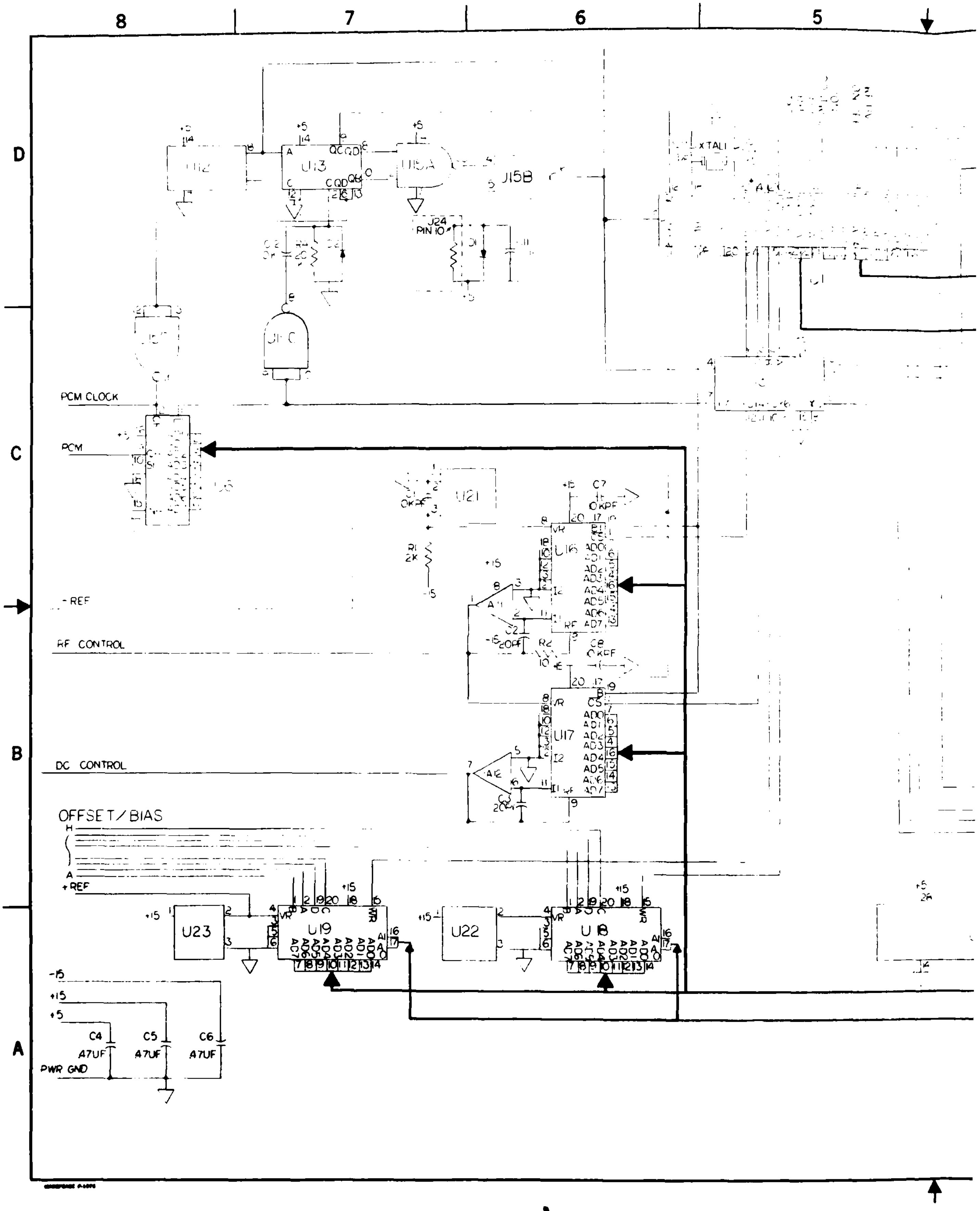
The codes specifying the flight time to nosecone ejection and the other events were stored in the EEPROM.

The binary code specifying the elapsed time between events was determined by taking the required number of seconds and multiplying that number by 10. The longest time interval to be specified was the time between the launch and the primary command to remove the nosecone. Therefore, two address locations in the EEPROM were assigned to time the primary command. The times for the back-up command and the command to activate the mass filter control circuits were referenced to the primary command. The number of seconds specified in the second byte were added to the time specified by the first byte. Thus, these two timed commands required only one memory location for each code.

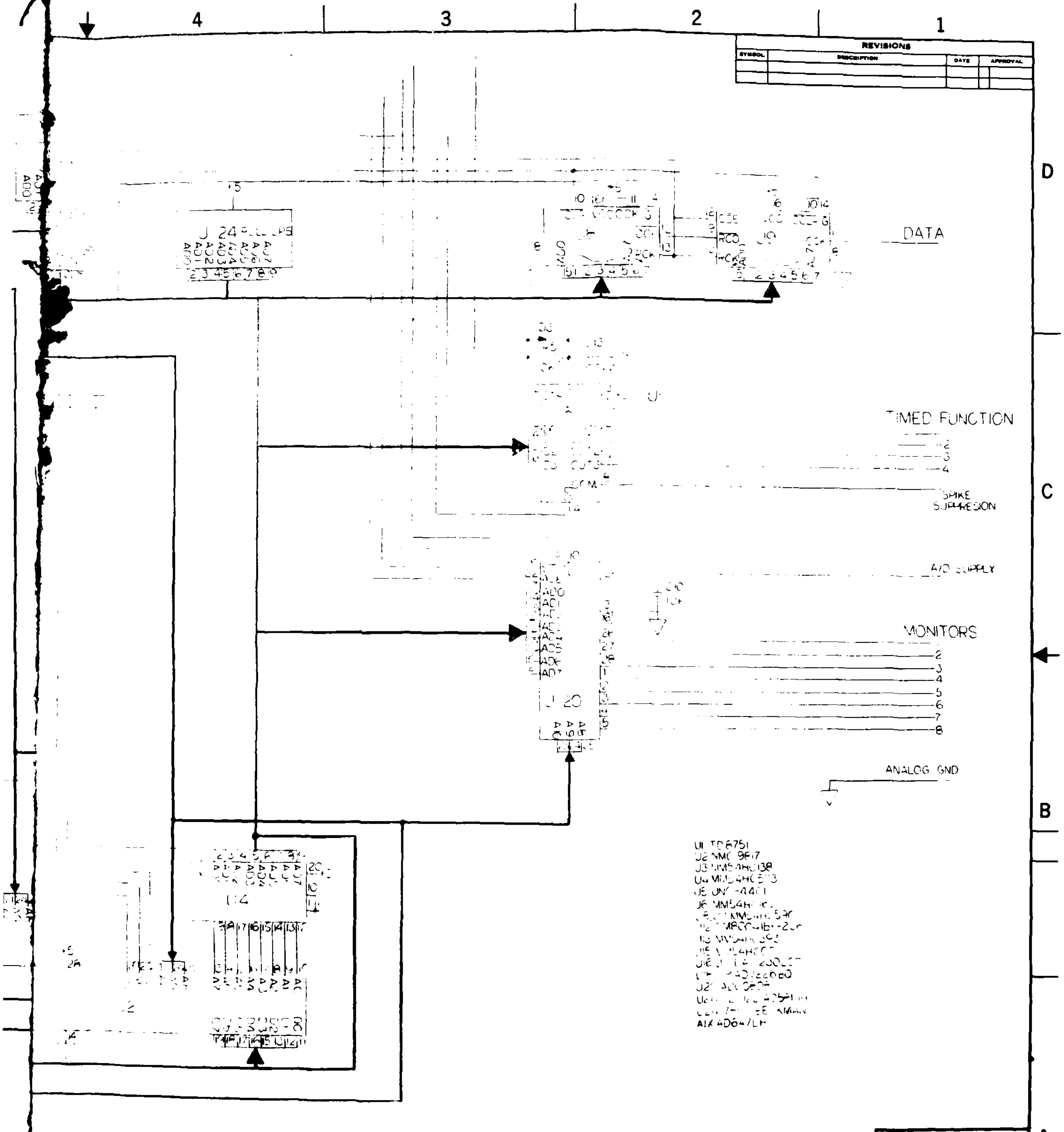
Until the time that the mass filter exciter and HV supplies were activated no meaningful data, except for the frame synchronization words (EB90H), were present within the PCM frame. Once the instrument was activated, the microcontroller loaded the mass filter control circuits with the first set of the stored parameters. The data counters were activated 1ms later. The timing for the operations continued to be provided by the interrupts from the PCM. The support and monitor data was gathered and stored in a bank of temporary storage registers for transmission during the next frame. The ion data was collected during an interval of 9ms in which 18 PCM data words were transmitted. The interval began with the second frame sync word and terminated with the onset of word 18. At that time the data was also transferred into two holding registers. The mass filter control parameter transfer into the circuits also was accomplished during word 18. Therefore, approximately 1ms of settling time was allowed before data gathering resumed with the filter set for a new atomic mass unit.

The mass filter control program was stored in the EEPROM. The first two address locations contained the frame synchronization words for the PCM data. The next four locations stored the time codes for the ejection of the nosecone and the activation of the mass filter. Remaining 10 address locations were left in reserve for other uses which could include an identification code and some other descriptive data for the stored program. The remainder of the 2 k byte EEPROM was reserved for the control program.

Sixteen locations were used to describe the parameters which stepped the filter through up to 3 atomic mass units. Common bias and ratio conditions were used for the operation in the three mass domains. When the present task was completed, the program advanced into the next block of 16 locations for new instructions. Thus, a total of 127 different parameter combinations could be stored before the flight program repeated. The control parameters and the sequence in which they were stored in the EEPORM are listed in Appendix C. The flow charts and the program of the operating system are presented in Appendix D and E respectively.



8
7
6
5
D
C
B
A
U12
U13
U15A
U15B
U11C
U15
J24 PIN 10
D1
D2
R4
XTAL1
U1
PCM CLOCK
PCM
U8
U21
R1 2K
C7 10KPF
U16
A11
C2 20PF
R2 10
C8 10KPF
U17
A12
C1 20PF
-REF
RF CONTROL
DC CONTROL
OFFSET/BIAS
H
A
+REF
U23
U19
U22
U18
VR
WR
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
CS
RF
A1
A0
-15
+15
+5
C4 47UF
C5 47UF
C6 47UF
PWR GND



NEXT ASSY	PROJECT
APPLICATION	

TOLERANCE UNLESS OTHERWISE NOTED:
DECIMAL: .XX ± .01
.XXX ± .005
FRACTIONAL: ± 1/64
ANGULAR: ± 0° 30'
SURFACE FINISH: 125✓
BREAK ALL SHARP EDGES AND BURRS
FINISH

DRAWN
CHECKED
SCALE
MATERIAL

ENG'R
DAVYS

DATE
1 JULY 65

LIPD
DIGITAL
BOARD

FIGURE 1

CONTRACT NUMBER 8513

NORTHEASTERN UNIVERSITY

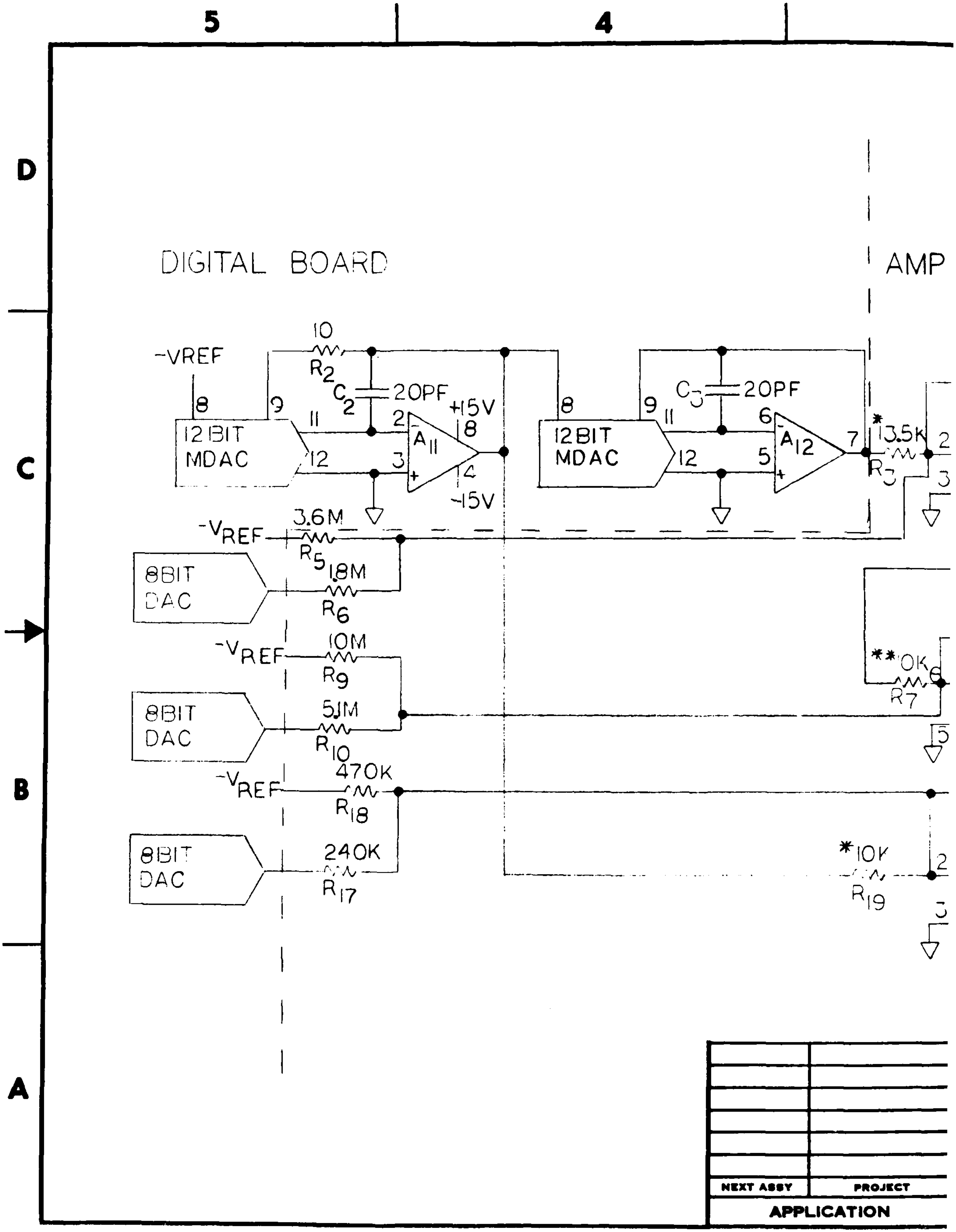
COLLEGE OF ENGINEERING

BOSTON, MASS. 02115

LIPD U002

A

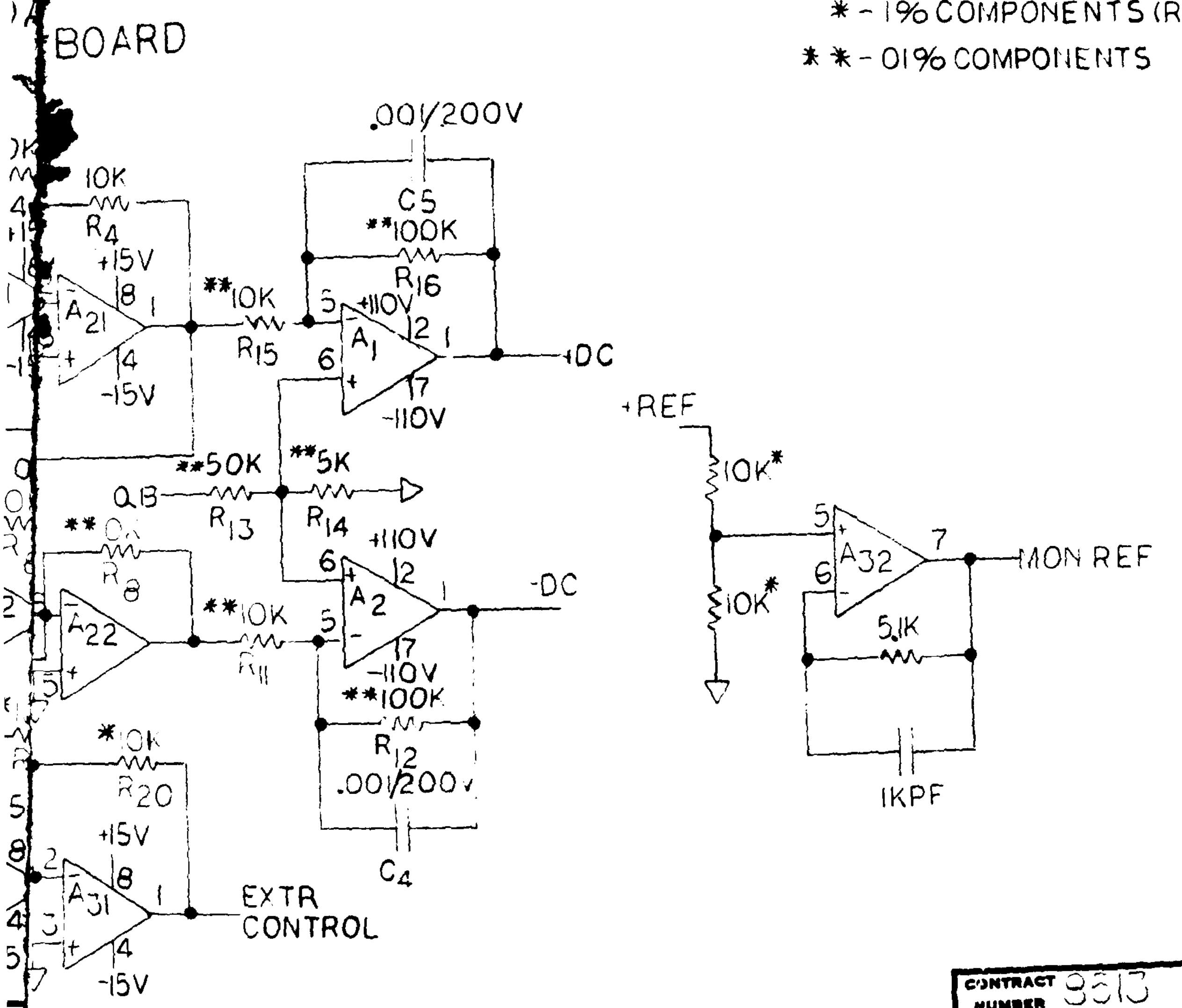
2



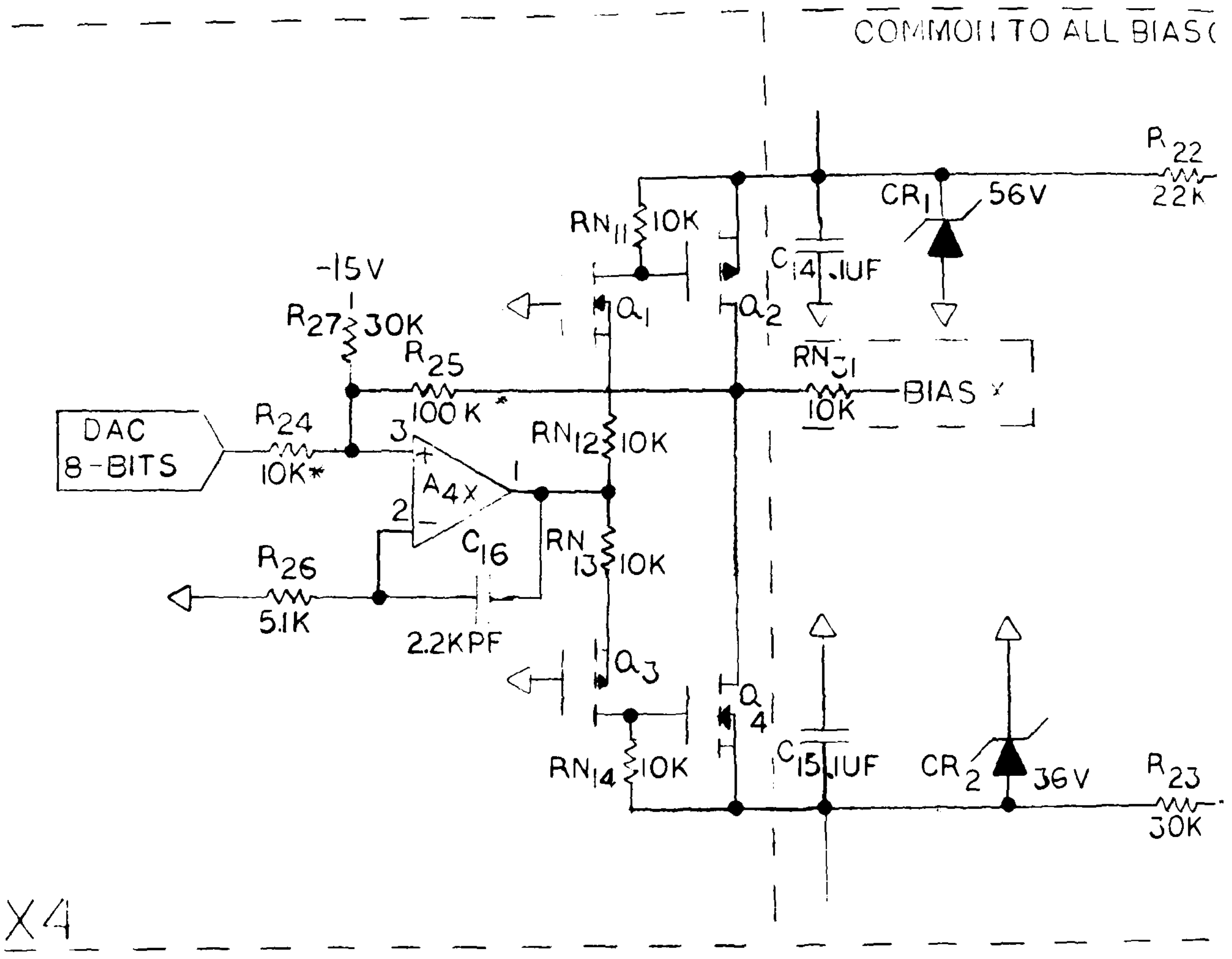
MAKEPEACE P-5600

REVISIONS			
SYMBOL	DESCRIPTION	DATE	APPROVAL

MDAC-DAC1230LCD
DAC-$\frac{1}{4}$ AD7226BQ
A_1-A_2-BB35823J
* - 1% COMPONENTS (RN55C)
** - 01% COMPONENTS



TOLERANCE UNLESS OTHERWISE NOTED:	DRAWN A.D	ENG'R R SUL S	DATE 10-3-85	CONTRACT NUMBER 3313
DECIMAL: x x ± .01 x x x ± .005	CHECKED			NORTHEASTERN UNIVERSITY
FRACTIONAL: ± 1/64 ANGULAR: ± 0° 30'	SCALE	LIPT EXCITER CONTROL		COLLEGE OF ENGINEERING BOSTON, MASS. 02115
SURFACE FINISH: 125	MATERIAL			
BREAK ALL SHARP EDGES AND DEBURR				LIPDAM000
FINISH		FIGURE 2		



NEXT ASSY	PROJECT
APPLICATION	

MAKEPEACE P-8600

REVISIONS			
SYMBOL	DESCRIPTION	DATE	APPROVAL

CIRCUITS

-110V

-110V

Q_1	-IRFD113
Q_2	-IRFD9122
Q_3	-IRFD 9123
Q_4	-IRFD 112
A-$\frac{1}{4}$	HA4602
DAC-$\frac{1}{4}$	AD7226BQ
C_1-	CKR06BX104K
C_2-	CKR06BX104K
C_3-	CKR105BX103K
C_4-	CKR105BX103K
✱-	RN55C(X2)
R's-	RC07(X7)
Z_1	IN4758
Z_2	IN4753
R_4-R_5-	898-3-10K BECKMAN

TOLERANCE UNLESS OTHERWISE NOTED: DECIMAL: x x ± .01 x x x ± .005 FRACTIONAL: ± 1/64 ANGULAR: ± 0° 30' SURFACE FINISH: 125√ BREAK ALL SHARP EDGES AND DEBURR	DRAWN ASO CHECKED SCALE MATERIAL	ENG'R RSUKYS LIPD BIAS CKT FIGURE 3	DATE 10-1-85
FINISH			

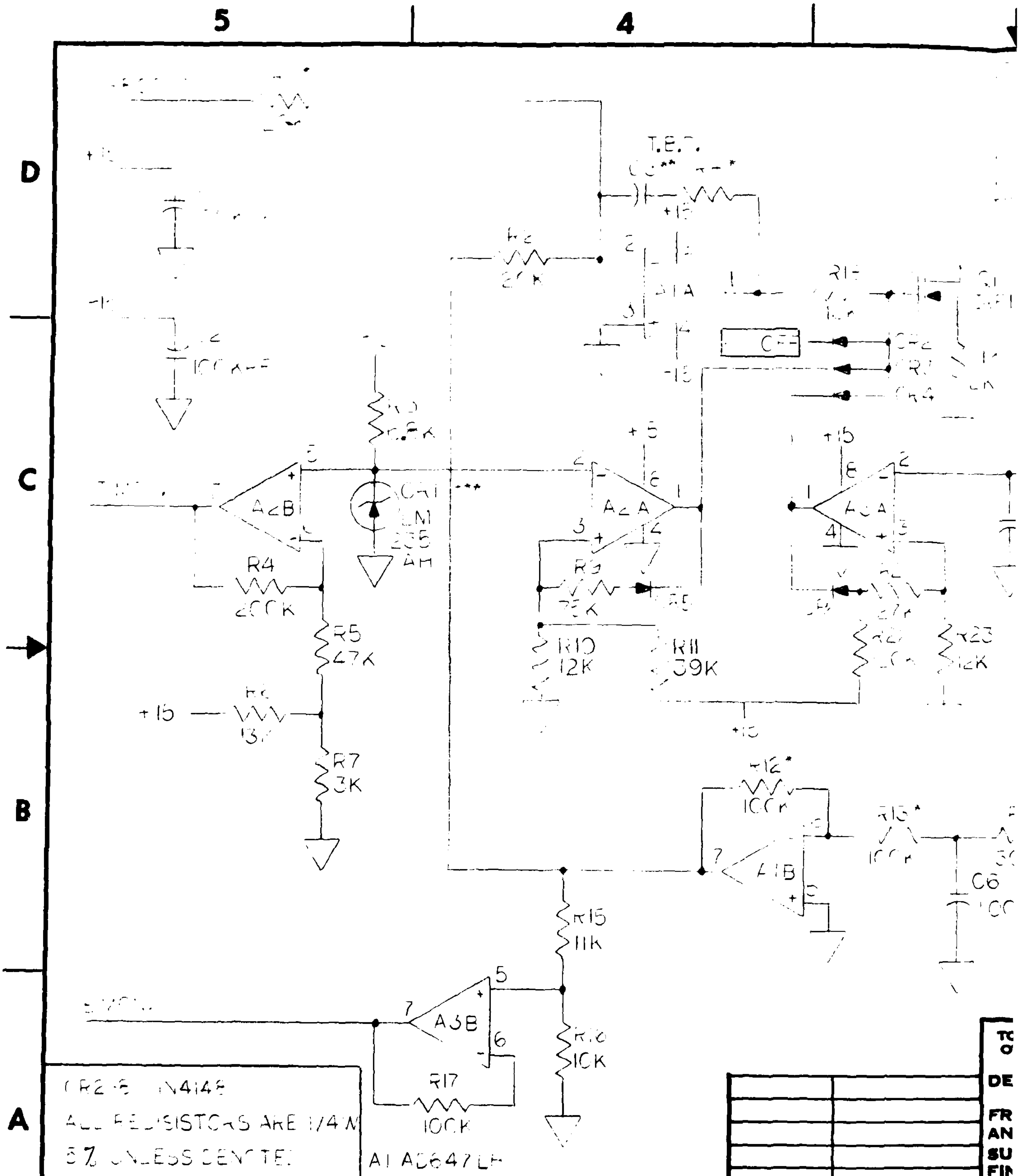
CONTRACT NUMBER 8513

NORTHEASTERN UNIVERSITY

COLLEGE OF ENGINEERING

BOSTON, MASS. 02115

LIPDB001



CR2-6 1N4148

ALL RESISTORS ARE 1/4 W

5% UNLESS DENOTED

* RN55C

** [illegible]

A1 AD647LH

A2-3 LM158

NEXT ASSY	PROJECT
APPLICATION	

TO
O

DE

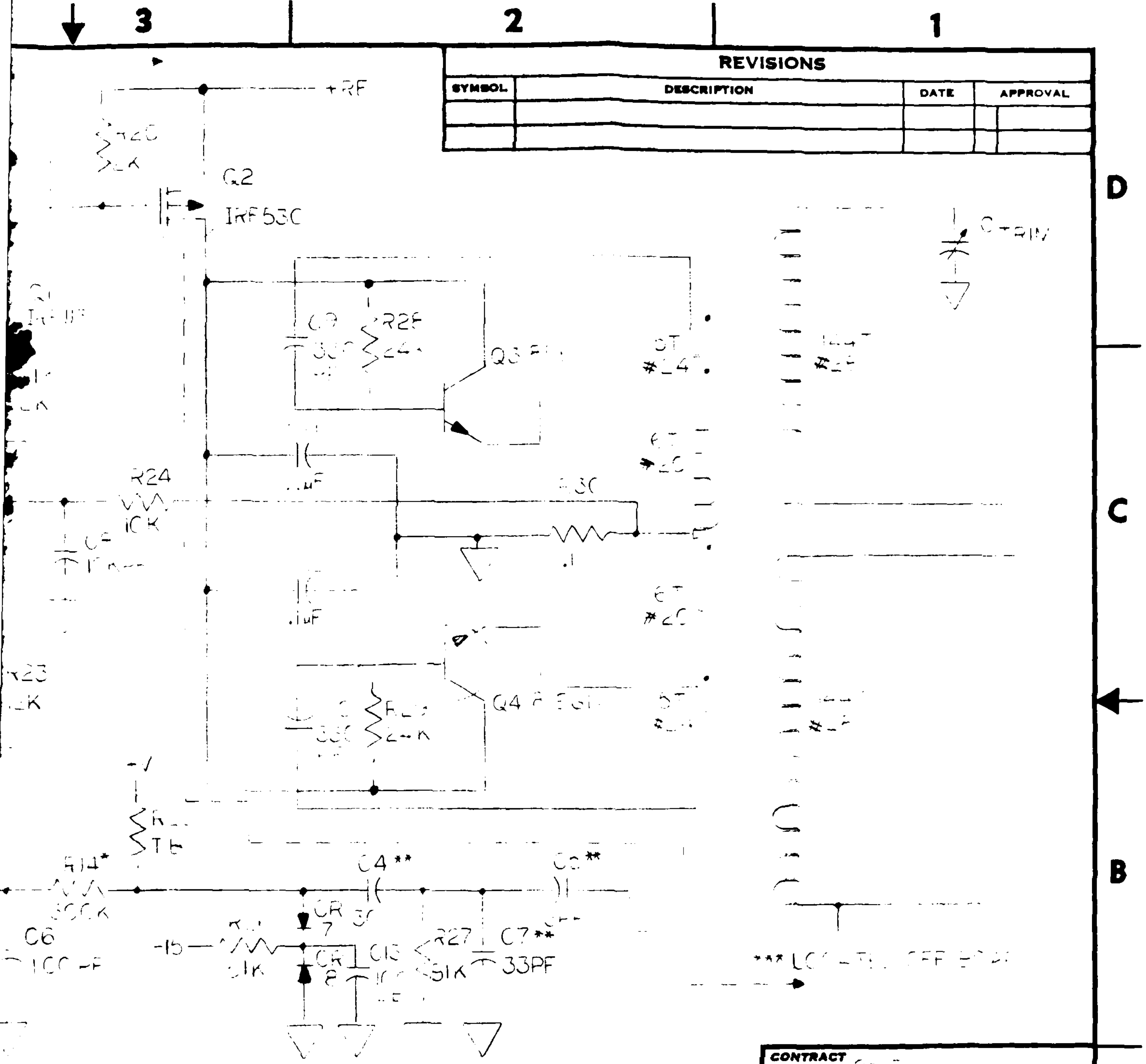
FR
AN

SU
FIN

BRE
AND

FINI

MAKEPEACE P-5600



REVISIONS

SYMBOL	DESCRIPTION	DATE	APPROVAL

TOLERANCE UNLESS OTHERWISE NOTED: DECIMAL: x x ± .01 x x x ± .005 FRACTIONAL: ± 1/64 ANGULAR: ± 0° 30' SURFACE FINISH: 125√ BREAK ALL SHARP EDGES AND DEBURR	DRAWN CHECKED SCALE MATERIAL	ENG'R R SUKYS	DATE 20 JUNE
FINISH		FIGURE 2	

CONTRACT NUMBER

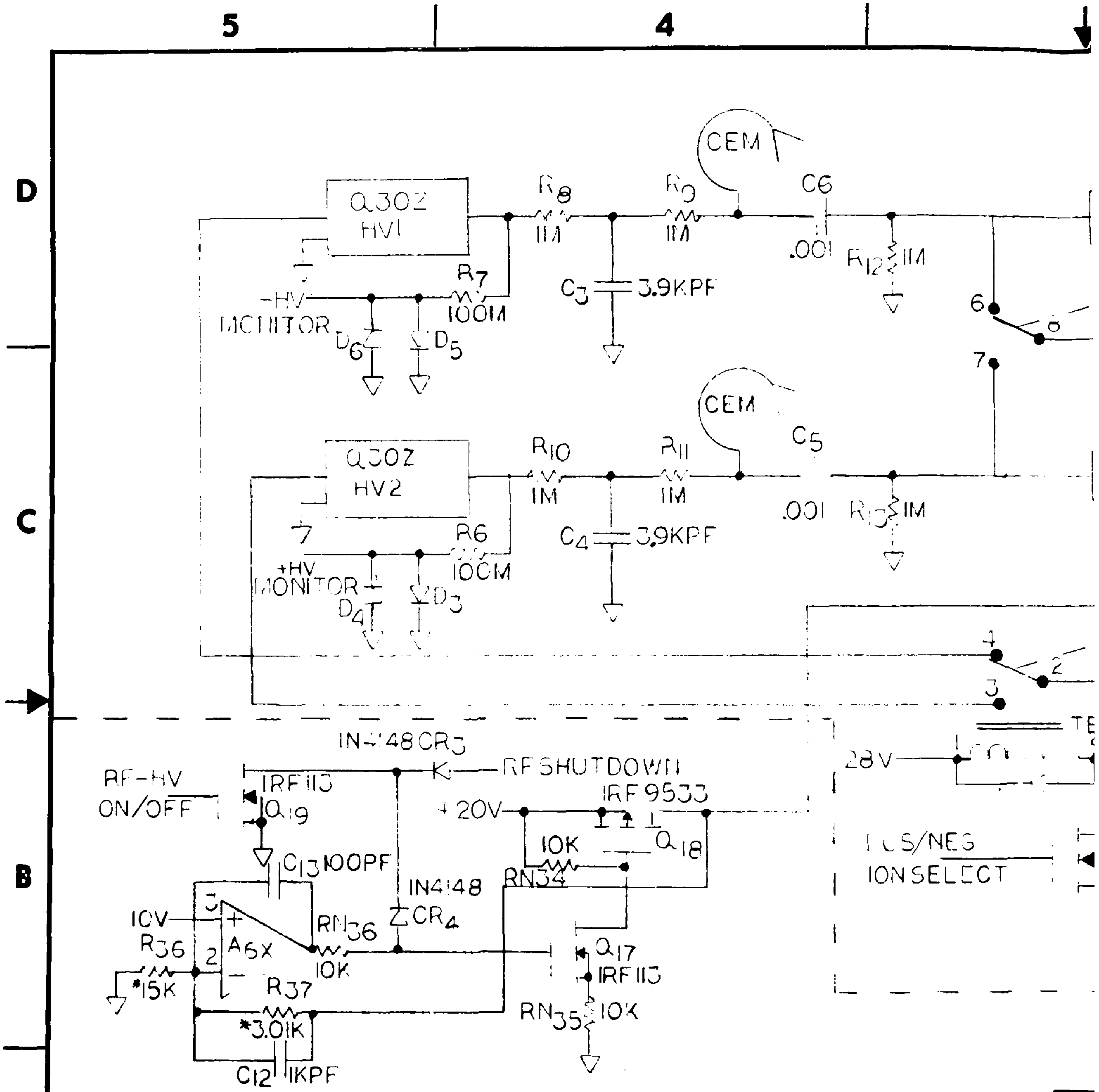
NORTHEASTERN UNIVERSITY

COLLEGE OF ENGINEERING

BOSTON, MASS. 02115

1

2



A6X LM647LH
DX- IN4148

A

NEXT ASSY	PROJECT
APPLICATION	

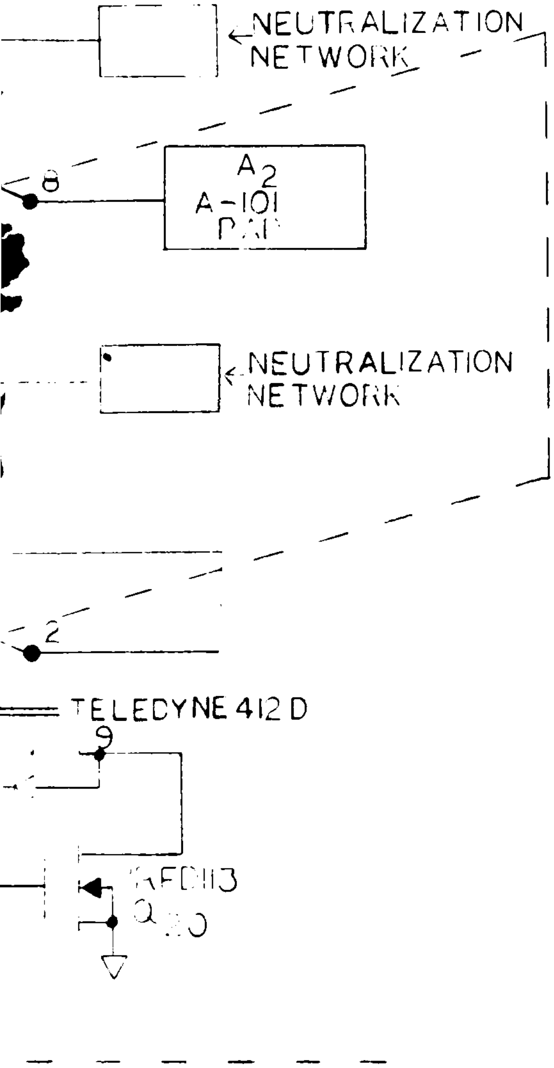
TO
OT
DE
FRA
AN
SU
FIN
BRE
AND
FINI

MAKEPEACE P-5600

3 2 1

REVISIONS			
SYMBOL	DESCRIPTION	DATE	APPROVAL

D



C

B

A

CONTRACT NUMBER 9510

TOLERANCE UNLESS OTHERWISE NOTED:

DECIMAL: × × ± .01
× × × ± .005
FRACTIONAL: ± 1/64
ANGULAR: ± 0° 30'
SURFACE FINISH: 125

BREAK ALL SHARP EDGES AND DEBURR

FINISH

DRAWN ASC

CHECKED

SCALE

MATERIAL

ENG'R R UKYS

DATE 10-2 8

EPD
HV SUPPLY
AND
CHARGE AMPLIFIER
CKTS
FIGURE 5

NORTHEASTERN UNIVERSITY

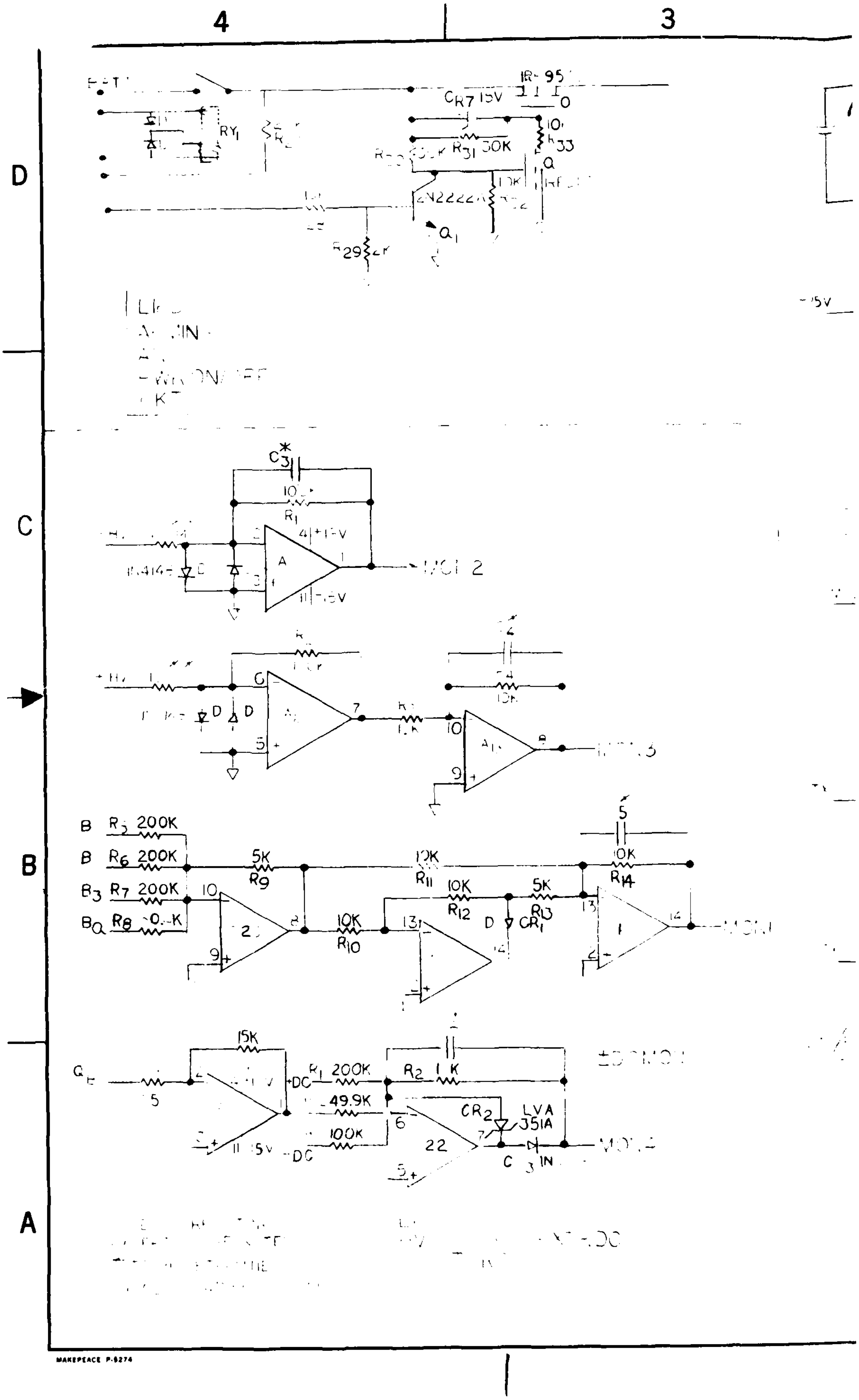
COLLEGE OF ENGINEERING

BOSTON, MASS. 02115

EPI-V00C

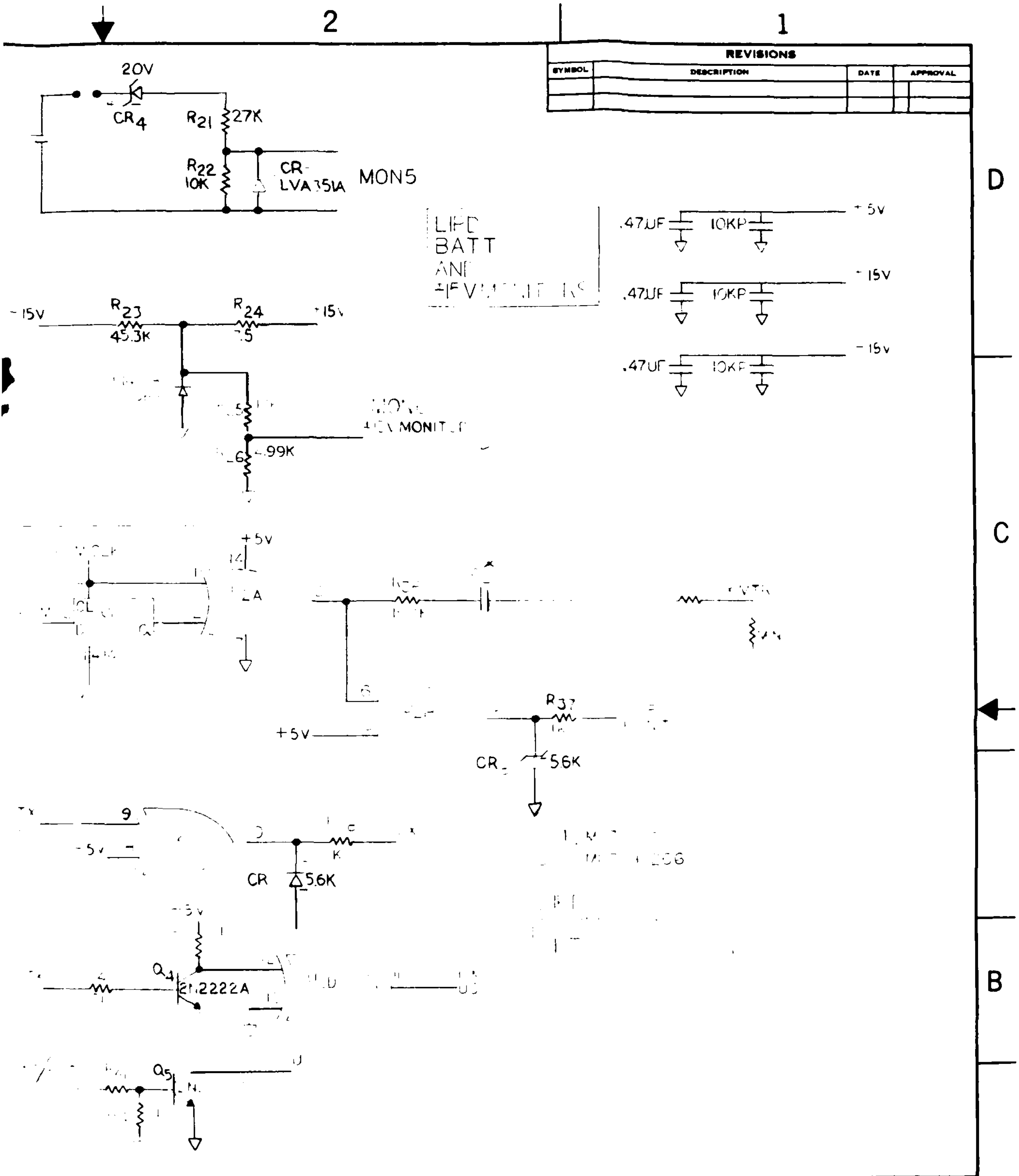
/

2



MAKEPEACE P-6274

REVISIONS			
SYMBOL	DESCRIPTION	DATE	APPROVAL



NEXT ASSY	PROJECT
APPLICATION	

TOLERANCE UNLESS OTHERWISE NOTED:
DECIMAL: x x ± .01
x x x ± .005
FRACTIONAL: ± 1/64
ANGULAR: ± 0° 30'
SURFACE FINISH: 125 ✓
BREAK ALL SHARP EDGES AND DEBURR

FINISH

DRAWN

CHECKED

SCALE

MATERIAL

ENG'R
UKYS

DATE

LIPD
MONITO
D

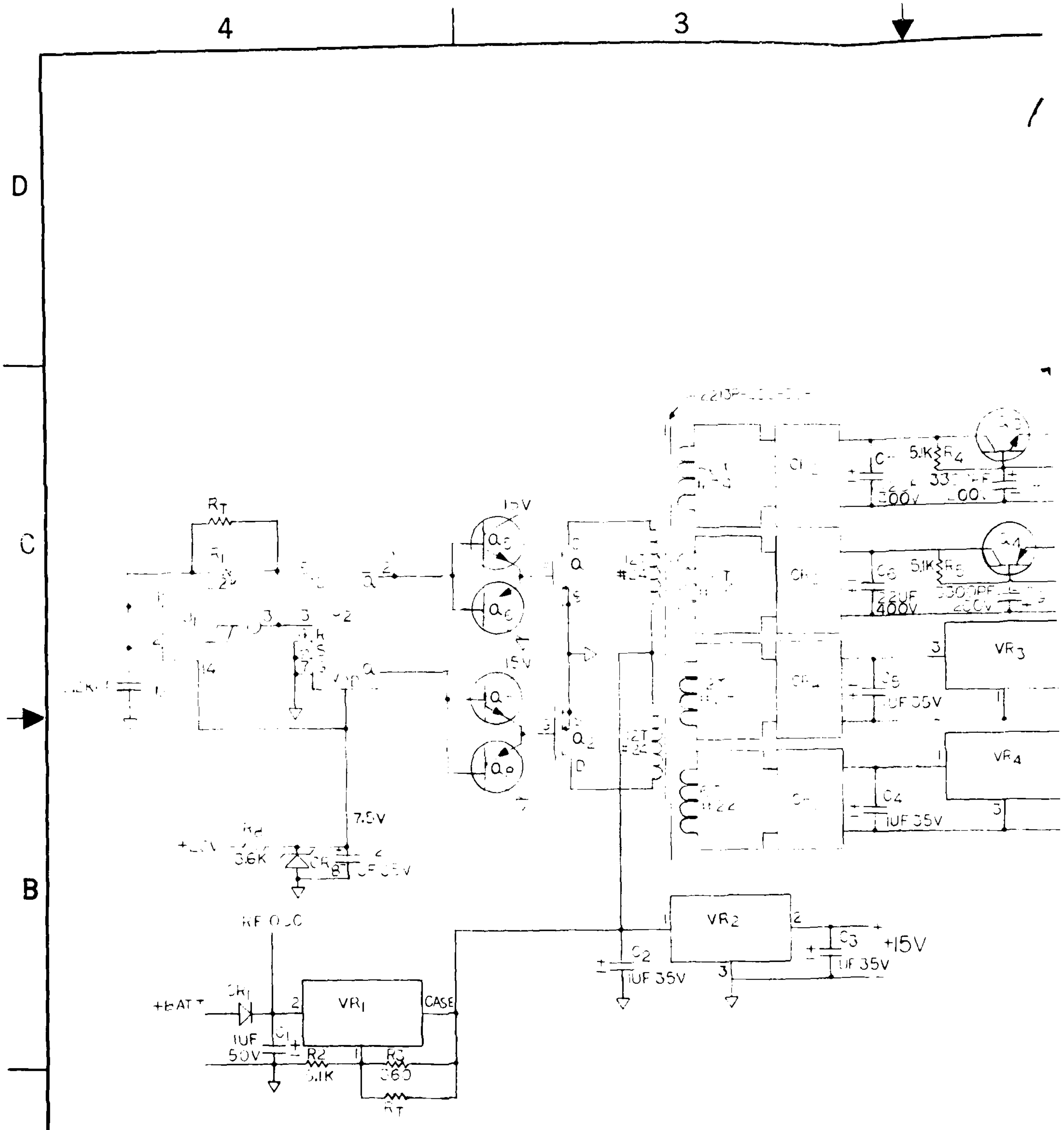
CONTRACT NUMBER 513

A

NORTHEASTERN UNIVERSITY

COLLEGE OF ENGINEERING
BOSTON, MASS. 02115

00



NEXT ASSY	PROJECT
APPLICATION	

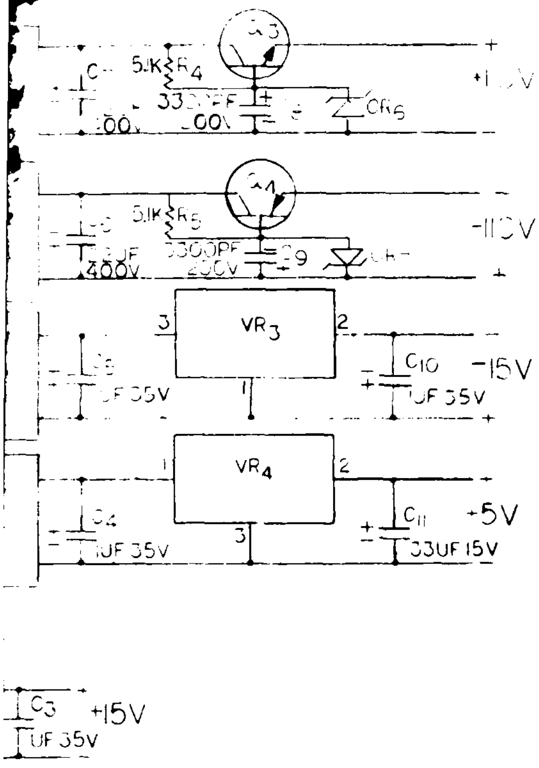
MAKEPEACE P-8274

2

1

REVISIONS			
SYMBOL	DESCRIPTION	DATE	APPROVAL

D



U_1-CD40C-3BF (1/2)
U_2-CD40-3BD (1/2)

Q_1-IRF 530
Q_2-IRF 730
Q_3-MJ7015
Q_4-MJ4035
Q_5-2N2222A
Q_6-2N2907A
Q_7-2N2222A
Q_8-2N2907A

CR_1-1N5814
CR_2-676-2
CR_3-670-2
CR_4-676-2
CR_5-676-2
CR_6-IN4490
CR_7-IN4490
CR_8-IN775

VR_1-LM217K
VR_2-UA78M15K
VR_3-UA79M15HC
VR_4-UA7805UC

C

B

A

CONTRACT NUMBER 8531

NORTHEASTERN UNIVERSITY

COLLEGE OF ENGINEERING
BOSTON, MASS. 02115

TOLERANCE UNLESS OTHERWISE NOTED:
DECIMAL: x x ± .01
x x x ± .005
FRACTIONAL: ± 1/64
ANGULAR: ± 0° 30'
SURFACE FINISH: 125√
BREAK ALL SHARP EDGES AND DEBURR

DRAWN ASO
ENG'R R SUKYS
DATE 8-19-83
CHECKED
SCALE
MATERIAL
FINISH

LIPD
POWER
SUPPLY

FIGURE 7

LIPD PS000

NEXT ASSY | PROJECT
APPLICATION

APPENDIX A

PCM FRAME

WORD 1	AMU CONTROL DATA MSBYTE
WORD 2	AMU CONTROL DATA LSBYTE
WORD 3	RATIO CONTROL DATA MSBYTE
WORD 4	RATIO CONTROL DATA LSBYTE
WORD 5	SPECTRA COUNTER DATA MSBYTE
WORD 6	SPECTRA COUNTER DATA LSBYTE
WORD 7	AC MONITOR
WORD 8	HV1 MONITOR
WORD 9	HV2 MONITOR
WORD 10	COMBINED BIAS MONITOR
WORD 11	± DC MONITOR
WORD 12	± 15V MONITOR
WORD 13	BATTERY V. MONITOR
WORD 14	TEMPERATURE MONITOR
WORD 15	QUADRUPOLE BIAS CONTROL DATA
WORD 16	BIAS 2 CONTROL DATA
WORD 17	BIAS 3 CONTROL DATA
WORD 18	BIAS 4 CONTROL DATA
WORD 19	FRAME SYNC WORD 1
WORD 20	FRAME SYNC WORD 2

APPENDIX B

COMMAND CODES

Command codes are given in the decimal notation. The address (A) and the data (D) represent hexadecimal numbers.

CMD 1.	128; AAA; DD	-	Enters data into the EEPROM.
CMD 2.	129; AAA	-	Sends data from the EEPROM.
CMD 3.	130; AA	-	Sends data from a selected monitor.
CMD 4.	131; DDD	-	Enters data into the Control DAC.
CMD 5.	132; DDD	-	Enters data into the Ratio DAC.
CMD 6.	133; AA; DD	-	Enters data into the selected Bias DAC.
CMD 7.	134; AA; DD	-	Enter data into the selected Offset DAC.
CMD 8.	135; AA	-	Sends data from the RAM.
CMD 9.	136; AA; DD	-	Enters data into the RAM.
CMD 10.	137;	-	RF/HV ON
CMD 11.	138	-	RF/HV OFF
CMD 12.	139	-	Positive Ion Mode
CMD 13.	140	-	Negative Ion Mode
CMD 14.	141; AAA	-	Executes a segment of a mass filter program and sends one to three frames of data through the serial link.
CMD 15.	142	-	Transfers control to the flight program.
CMD 16.	168	-	End of transmission code.
CMD 17.	127	-	Reset
CMD 18.	255	-	This code is sent back to the external control to indicate an error in the received instruction.

COMMAND ADDRESS ASSIGNMENTS

CMD 1 & 2	EEPROM	000H-7FFH
CMD 3	MONITORS:	
	1. COMBINED BIAS	00H
	2. ± 15V	01H
	3. HV1	02H
	4. HV2	03H
	5. ± DC	04H
	6. BATTERY	05H
	7. AC AMPLITUDE	06H
	8. TEMPERATURE	07H
CMD 6	BIAS:	
	1. DAC A (QUADRUPOLE)	00H
	2. DAC B	01H
	3. DAC C	02H
	4. DAC D	03H
CMD 7	OFFSET:	
	1. DAC A (± DC)	00H
	2. DAC B (-DC)	01H
	3. DAC C (AC)	02H
CMD 8 & 9	RAM	00-7FH
CMD 14	EEPROM PROGRAM BLOC	(010 X N) H WHERE 01H ≤ N ≤ 7FH

APPENDIX C

EEPROM DATA FORMAT

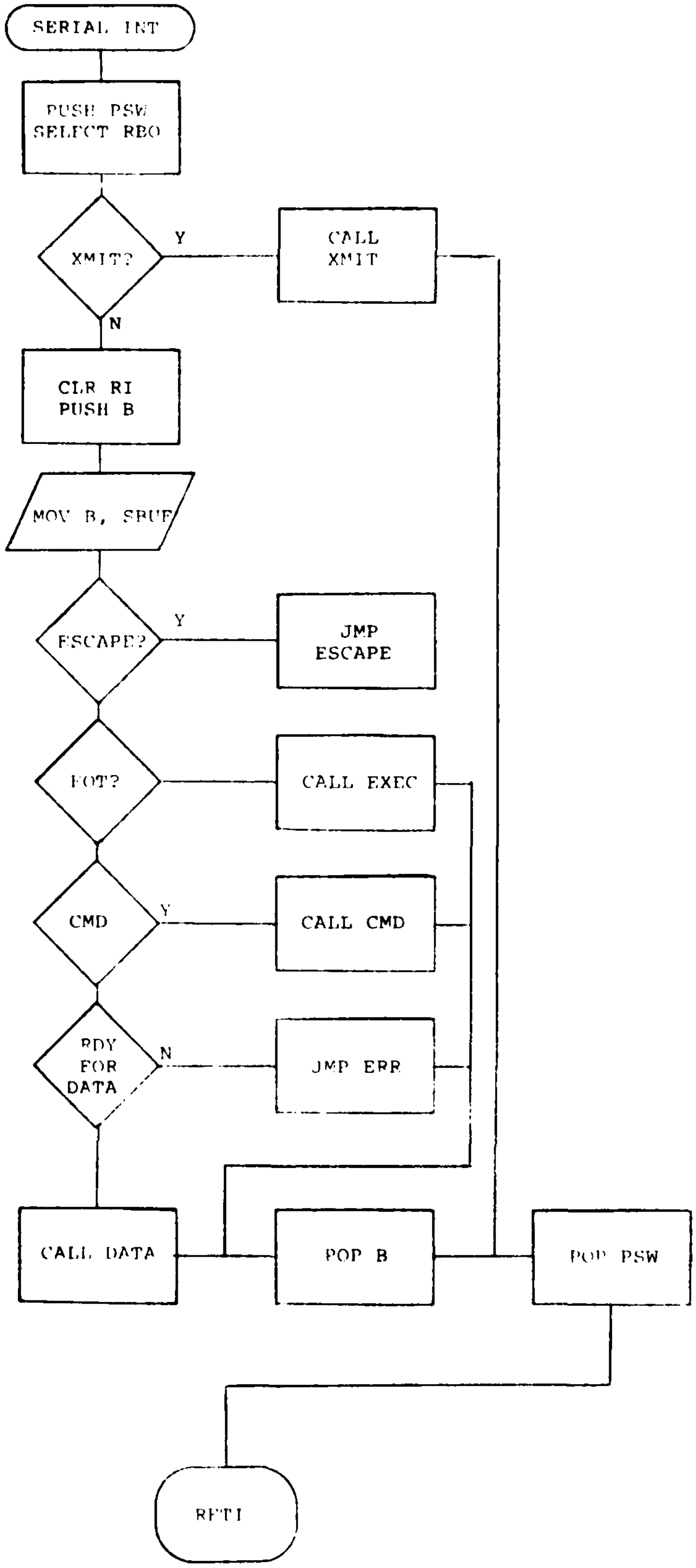
000H	FRAME SYNC. WORD 1 (EBH)
001H	FRAME SYNC. WORD 2 (90H)
002H	NOSECONE EJECT TIME: FIRST INTERVAL
003H	+ SECOND INTERVAL
004H	BACK-UP NOSECONE EJECT TIME
005H	HV AND AC EXCITER ON TIME
006H TO 00FH	AVAILABLE FOR COMMENTS
XX0H	HI-BYTE CONTROL DAC (1st AMU)
XX1H	LO-BYTE CONTROL DAC
XX2H	HI-BYTE RATIO DAC
XX3H	LO-BYTE RATIO DAC
XX4H	OFFSET DAC A
XX5H	OFFSET DAC B
XX6H	OFFSET DAC C
XX7H	BIAS DAC A (QUADRUPOLE)
XX8H	BIAS DAC B
XX9H	BIAS DAC C
XXAH	BIAS DAC D
XXBH	HI-BYTE CONTROL DAC (2nd. AMU)
XXCH	LO-BYTE CONTROL DAC
XXDH	HI-BYTE CONTROL DAC (3rd. AMU)
XXEH	LO-BYTE CONTROL DAC
XXFH	END OF PAGE/PROGRAM FLAG (00H/FFH)

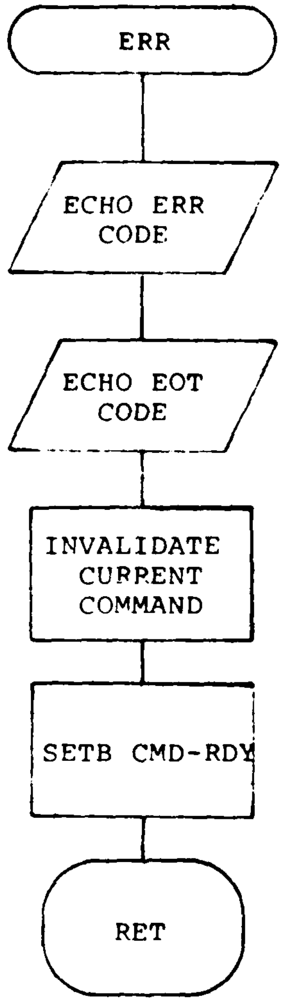
<u>NOTE</u>:

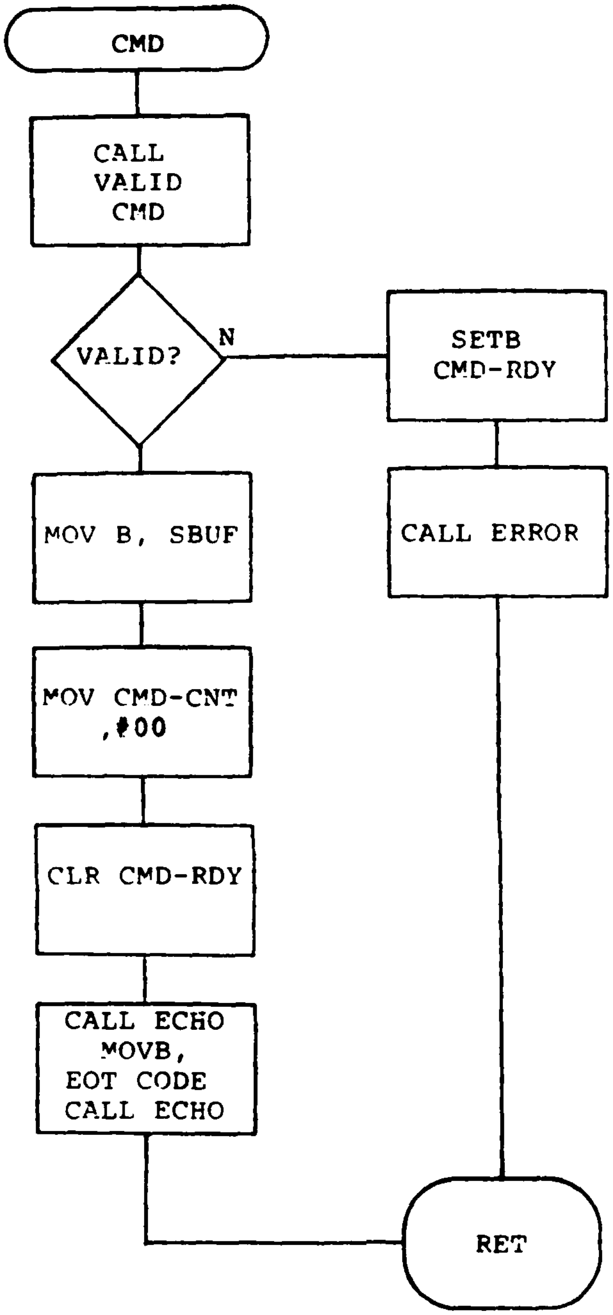
1. Control and ratio DAC data 12 bits left justified.
2. 00H in locations XXBH, XXDH and XXFH advances the program to the next page. [p. XX0H to p. (XX+1) 0H].
3. FFH in locations XXBH, XXDH and XXFH returns program to the first page [p.XX0H to p.010H].

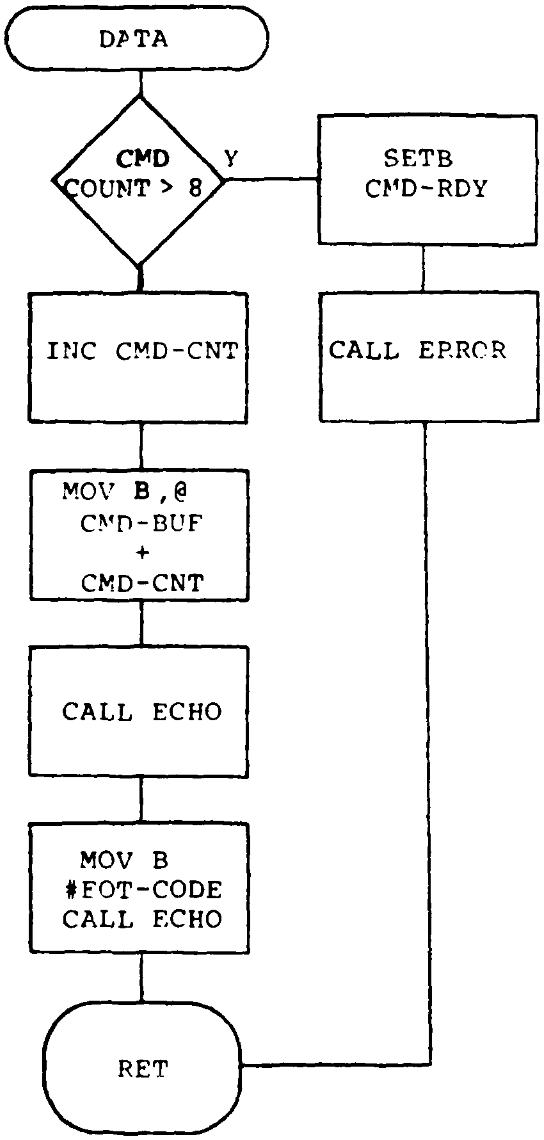
APPENDIX D

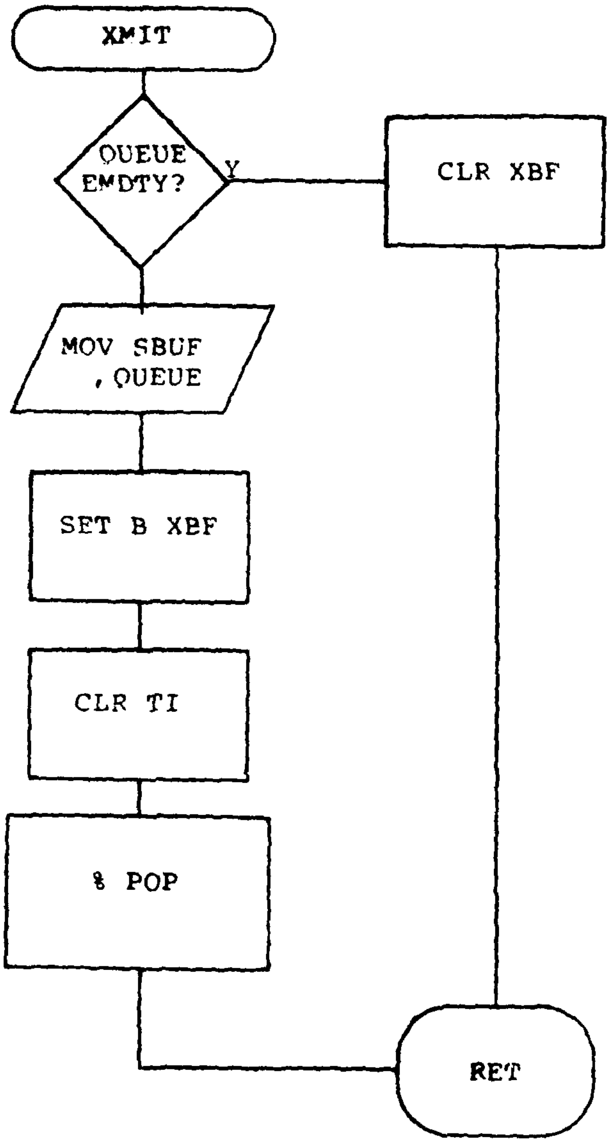
FLOW GRAPHS

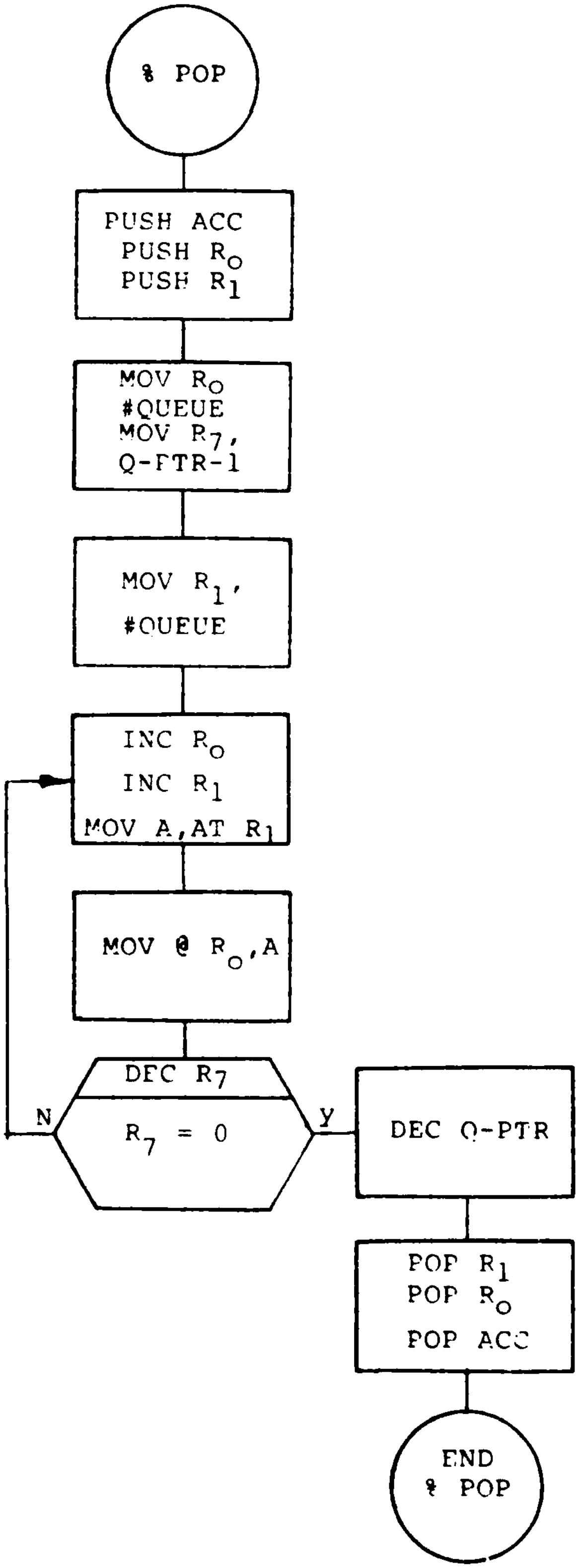

SERIAL INT
PUSH PSW
SELECT RBO
XMIT?
Y
CALL XMIT
N
CLR RI
PUSH B
MOV B, SBUF
ESCAPE?
Y
JMP ESCAPE
EOT?
CALL EXEC
CMD
Y
CALL CMD
RDY
FOR
DATA
N
JMP ERR
CALL DATA
POP B
POP PSW
RETI

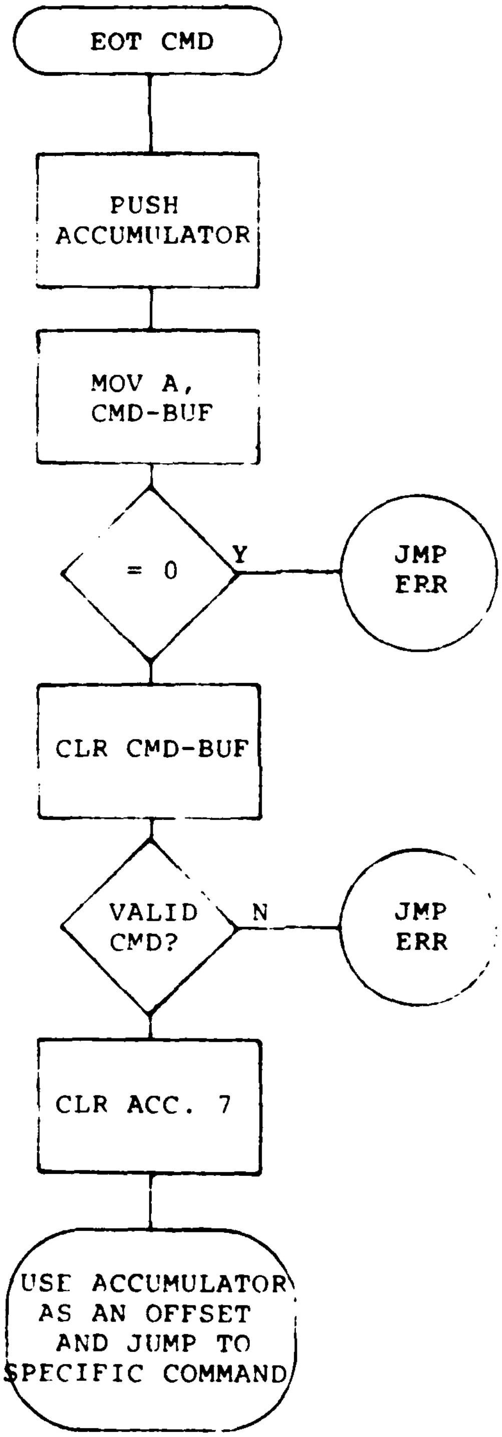

ERR
ECHO ERR CODE
ECHO EOT CODE
INVALIDATE CURRENT COMMAND
SETB CMD-RDY
RET

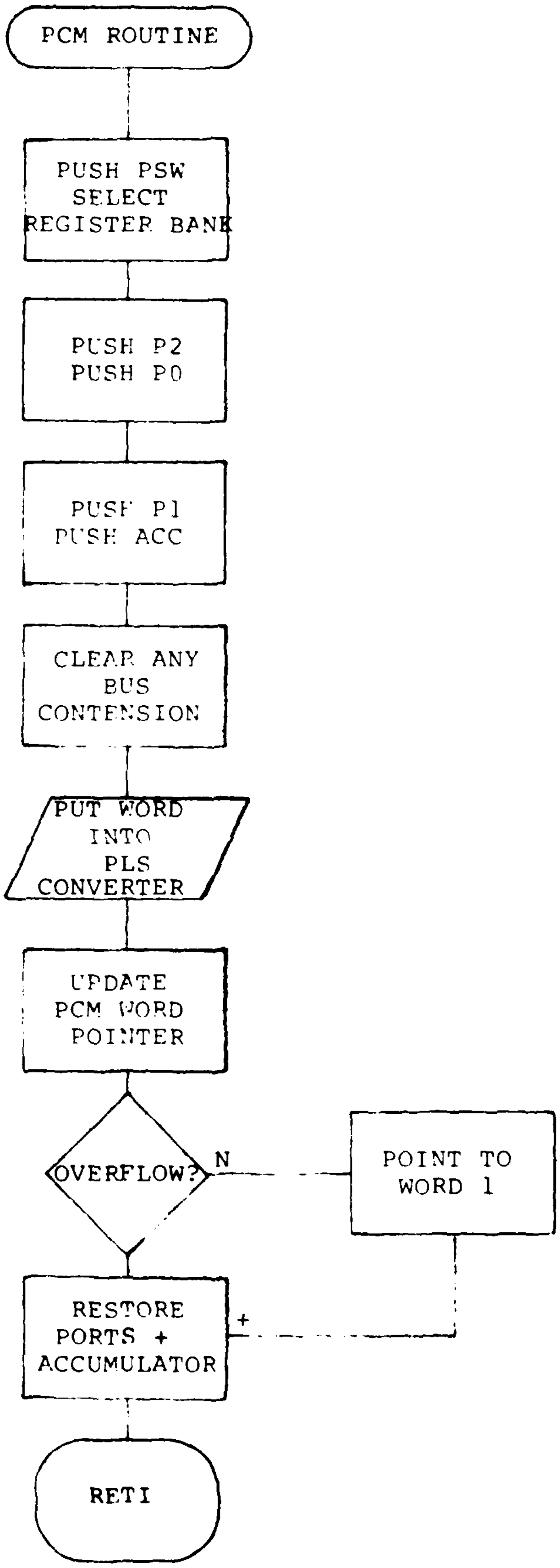

CMD
CALL VALID CMD
VALID?
N
SETB CMD-RDY
CALL ERROR
MOV B, SBUF
MOV CMD-CNT ,#00
CLR CMD-RDY
CALL ECHO MOVB, EOT CODE CALL ECHO
RET

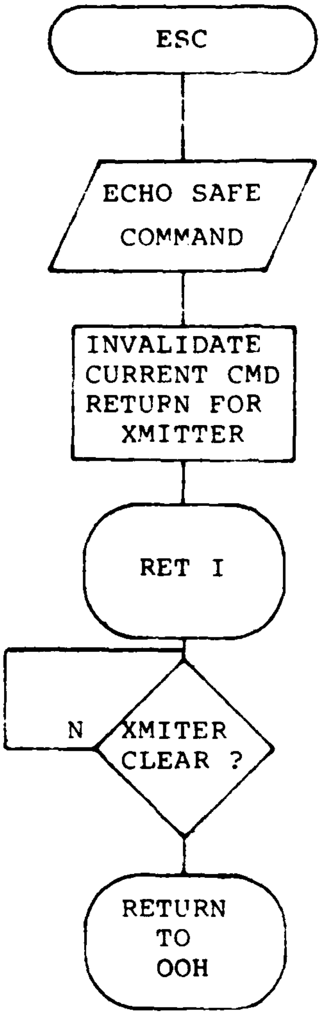

DATA
CMD COUNT > 8
Y
SETB CMD-RDY
INC CMD-CNT
CALL ERROR
MOV B,@ CMD-BUF + CMD-CNT
CALL ECHO
MOV B #EOT-CODE CALL ECHO
RET


XMIT
QUEUE EMDTY?
Y
CLR XBF
MOV SBUF ,QUEUE
SET B XBF
CLR TI
% POP
RET


% POP
PUSH ACC
PUSH R_O
PUSH R_1
MOV R_O
#QUEUE
MOV R_7,
Q-PTR-1
MOV R_1,
#QUEUE
INC R_O
INC R_1
MOV A,AT R_1
MOV @ R_O,A
DEC R_7
N
R_7 = 0
y
DEC Q-PTR
POP R_1
POP R_O
POP ACC
END
% POP


EOT CMD
PUSH ACCUMULATOR
MOV A, CMD-BUF
= 0
Y
JMP ERR
CLR CMD-BUF
VALID CMD?
N
JMP ERR
CLR ACC. 7
USE ACCUMULATOR AS AN OFFSET AND JUMP TO SPECIFIC COMMAND


PCM ROUTINE
PUSH PSW
SELECT
REGISTER BANK
PUSH P2
PUSH P0
PUSH P1
PUSH ACC
CLEAR ANY
BUS
CONTENSION
PUT WORD
INTO
PLS
CONVERTER
UPDATE
PCM WORD
POINTER
OVERFLOW?
N
POINT TO
WORD 1
+
RESTORE
PORTS +
ACCUMULATOR
RETI


ESC
ECHO SAFE COMMAND
INVALIDATE CURRENT CMD RETURN FOR XMITTER
RET I
N
XMITER CLEAR ?
RETURN TO OOH

APPENDIX E

PROGRAM

```
ISIS-II MCS-51 MACRO ASSEMBLER V2.1
OBJECT MODULE PLACED IN :F1:LIPD.OBJ
ASSEMBLER INVOKED BY:  :F1:ASM51 :F1:LIPD.ASM


LOC  OBJ            LINE     SOURCE

                       1     $PL(50)
                       2     $TITLE(                LIPD (SUPER ARCAS))
                       3     $XREF
                       4     $DATE(16 OCT 85)
                       5     $SB
                       6     $DEBUG
                       7     $EP(:F1:LIPD.ERR)
                       8     $NOGE
                       9     $INCLUDE  :F1:LIPD.DEF
                 =1   10     ;#####################################################################
                 =1   11     ;#                                                                   #
                 =1   12     ;#                    LIPD DEFINITION SECTION                        #
                 =1   13     ;#                                                                   #
                 =1   14     ;#####################################################################
                 =1   15     ;
                 =1   16     ;
0008             =1   17     PCM_POINT EQU   08H     ;R0 RB1 USE AS POINTER IN QUEUE FOR PCM WORD
----             =1   18     DSEG AT 09H
                 =1   19     ; RB 1 R1-R7 AND RB2 + RB3 ARE NOT USED SO ASSIGN AS MEMORY
0009             =1   20     AMU_H:  DS      01      ;PCM WORD 2
000A             =1   21     AMU_L:  DS      01      :         3
000B             =1   22     RAT_H:  DS      01      :         4
000C             =1   23     RAT_L:  DS      01      :         5
000D             =1   24     Q_BIAS: DS      01      ;COM Q BIAS BIAS DAC 0 #0 ...PCM WORD 16
000E             =1   25     BIAS_1: DS      01      ;WORD 17
000F             =1   26     BIAS_2: DS      01      ;WORD 18
0010             =1   27     BIAS_3: DS      01      ;WORD 19
0011             =1   28     CAP_SAFETY:     DS      01
                 =1   29     ;
                 =1   30     ;
----             =1   31     DSEG AT 30H
                 =1   32     ;
0030             =1   33     CMD_BUF:        DS      8       ;SAVE 8 BYTES FOR COMMAND DECODING
0038             =1   34     CMD_CNT:        DS      01      ;# BYTES OF DATA RECIEVED FOR COMMAND
0039             =1   35     QUEUE:          DS      16H     ;20 BYTES FOR PCM/DATA/ECHO QUEUE
004F             =1   36     Q_PTR:          DS      01              ;HOLD # OF BYTS LEFT IN SERIAL QUEUE
0050             =1   37     STACK:          DS      01              ;AND THE REMAINING FOR A STACK
                 =1   38     ;
                 =1   39     ;
----             =1   40     BSEG            ;DEFINE THE BIT FLAGS
```

```
LOC  OBJ         LINE     SOURCE

            =1    41      ;
 0010       =1    42      CMD_RDY        BIT     10H
 0011       =1    43      TBF            BIT     11H     ;XMIT BUSY FLAG
 0012       =1    44      CMD1FLG        BIT     12H
 0013       =1    45      IS_FLT         BIT     13H
 0014       =1    46      SERIAL_STEP    BIT     14H     ;COMMUNICATE DATA VIA THE SERIAL LINK FLAG
 0015       =1    47      MULTI_AMU      BIT     15H     ;MULTIPLE AMU FLAG
 0016       =1    48      TIME_BIT       BIT     16H     ;TOGGLING AT INT1 RATE
 0017       =1    49      SAVE_BIT       BIT     17H     ;STORAGE SPACE FOR MON OE DURING INT1
 0094       =1    50      MON_ALE        BIT     P1.4    ;MONITOR ALE PIN
 0093       =1    51      MON_STRT       BIT     P1.3    ;MONITOR START CONVERSION PIN
 0092       =1    52      MON_OE         BIT     P1.2    ;MONITOR OUTPUT ENABLE PIN
 0090       =1    53      XFR            BIT     P1.0    ;DC/RAT DAC TRANSFER PIN
 0097       =1    54      ION_CONT       BIT     P1.7    ;POSITIVE NEGATIVE ION CONTROL PIN
 00B5       =1    55      RF_HV          BIT     P3.5    ;RF HV DISABLE PIN
 00B4       =1    56      TST            BIT     P3.4    ;FLIGHT/TEST PIN
 0091       =1    57      CNT            BIT     P1.1    ;COUNTER LATCH/COUNT PIN
 00A6       =1    58      SELECT         BIT     P2.6    ;7 LINE DECODER ENABLE PIN
 0095       =1    59      STROBE         BIT     P1.5    ;RELAY DRIVER STROBE PIN
 0096       =1    60      OD             BIT     P1.6    ;RELAY DRIVER OUTPUT DISABLE (1 = OFF
            =1    61      ;
            =1    62      ;
            =1    63      ;************************************************************************
            =1    64      ;*                                                                      *
            =1    65      ;*                        SELECTOR VALUES FOR PORT 2                    *
            =1    66      ;*                                                                      *
            =1    67      ;************************************************************************
 0040       =1    68      EEPROM_SELECT  EQU     40H     ; X1000XXX      Y0
 0048       =1    69      D_HIGH         EQU     48H     ; X1001XXX      Y1
 0050       =1    70      D_LOW          EQU     50H     ; X1010XXX      Y2
 0058       =1    71      LSB_RF         EQU     58H     ; 01011XXX      Y3
 00D8       =1    72      MSB_RF         EQU     0D8H    ; 11011XXX      Y3
 0060       =1    73      LSB_DC         EQU     60H     ; 01110XXX      Y4
 00E0       =1    74      MSB_DC         EQU     0E0H    ; 11110XXX      Y4
 0068       =1    75      BIAS_0_SELECT  EQU     68H     ; X1101XXX      Y5
 0070       =1    76      BIAS_1_SELECT  EQU     70H     ; X1110XXX      Y6
 0078       =1    77      PCM_LOAD       EQU     78H     ; X1111XXX      Y7
            =1    78      ;
            =1    79      ;
            =1    80      ;
 0000       =1    81      MON_SELECT     EQU     00H
            =1    82      ;
            =1    83      ;
            =1    84      ;WORKING VALUES
            =1    85      ;
```

```
LOC  OBJ            LINE     SOURCE

  0041          =1    86     AA       EQU     'A'
  0047          =1    87     G        EQU     'G'
  0030          =1    88     ZER      EQU     '0'
  0039          =1    89     NIN      EQU     '9'
  0006          =1    90     RF_MON   EQU     06H
  0002          =1    91     HV_1_MON EQU     02H
  0003          =1    92     HV_2_MON EQU     03H
  0000          =1    93     COMB_MON EQU     00H
  0004          =1    94     DC_MON   EQU     04H
  0001          =1    95     VCC_MON EQU      01H
  0005          =1    96     BAT_MON EQU      05H
  0007          =1    97     TEMP_MON EQU     07H
  00F2          =1    98     BAUD_COUNT EQU  0F2H    ;1200 BAUD COUNT RATE
                =1    99     :
                =1   100     :
                =1   101     :
                =1   102     ;SPECIAL COMMANDS/CODES
                =1   103     :
  007F          =1   104     ESC_CODE        EQU     7FH
  00FF          =1   105     ERR_CODE        EQU     0FFH
  00AB          =1   106     EOT_CODE        EQU     0ABH
                =1   107     ;
                =1   108     :
                =1   109     :
----            =1   110     XSEG AT 4000
0FA0            =1   111     FRAME_WORD_0:   DS      1       ;FRAME SYNC WORD 0 AT EEPROM 000
0FA1            =1   112     FRAME_WORD_1:   DS      1       :                1          001
0FA2            =1   113     CAP_DELAY:      DS      1       ;THE DELAY COUNT OF INT0'S TO BLOW CAP
0FA3            =1   114     DELAY_2:        DS      1       ;THE SECONDARY DELAY TO REPEAT THE BLOWING
                =1   115     :                                THE CAP AND START OF THE PROFILE
0FA4            =1   116     DELAY_3:        DS      1
0FA5            =1   117     TRASH_EE:       DS      11      ; WORDS RESERVED FOR USER RECORDS
0FB0            =1   118     PROFILE:        DS      2032    ; THE REST OF THE EEPROM
                =1   119     :
                =1   120     :
                =1   121     $EJECT
```

```
LOC  OBJ            LINE     SOURCE

               =1    122     SERIAL_BOOT_CODE SEGMENT CODE
               =1    123     PROFILE_FLT_CODE SEGMENT CODE
               =1    124     $EJECT
```

```
LOC  OBJ            LINE     SOURCE

                     125     $INCLUDE(:F1:POP.MAC)
                =1   126     ;####################### LIFO MACRO DEFINITION SECTION #######################
----            =1   127     CSEG
                =1   128     %*DEFINE(POP)
                =1           LOCAL FIFO
                =1           LOCAL TO_DONE
                =1           (
                =1           %'............... FIRST IN FIRST OUT OF QUEUE .........................%'
                =1                   PUSH    00H
                =1                   PUSH    01H
                =1                   MOV     R0,#QUEUE
                =1                   MOV     R1,#QUEUE
                =1                   INC     R1
                =1                   MOV     R7,Q_PTR
                =1                   DJNZ    R7,%FIFO
                =1                   JMP     %TO_DONE
                =1           %FIFO:
                =1                   MOV     A,@R1
                =1                   MOV     @R0,A
                =1                   INC     R1
                =1                   INC     R0
                =1                   DJNZ    R7,%FIFO
                =1           %TO_DONE:
                =1                   MOV     @R1,#00H
                =1                   DEC     Q_PTR
                =1                   POP     01H
                =1                   POP     00H
                =1           )
                =1   129     %*DEFINE(WAIT(WAIT_TIME))
                =1           LOCAL TO_DO_IT
                =1           LOCAL WAIT_ONE_MS
                =1           LOCAL TO_DONE
                =1           (
                =1           %'................ WAIT WAIT_TIME/10 SECONDS ..........................%'
                =1                   PUSH    ACC
                =1                   PUSH    00H
                =1                   PUSH    01H
                =1                   MOV     R1,%WAIT_TIME
                =1           %TO_DO_IT:
                =1                   MOV     A,R1
                =1                   JZ      %TO_DONE
                =1                   MOV     R0,#100D
                =1                   SETB    TIME_BIT
                =1           %WAIT_ONE_MS:
                =1                   JB      TIME_BIT,$
```

```
LOC  OBJ          LINE    SOURCE

                =1                  JNB     TIME_BIT,$
                =1                  DJNZ    R0,%WAIT_ONE_MS
                =1                  DEC     R1
                =1                  JMP     %TO_DO_IT
                =1            %TO_DONE:
                =1                  POP     01H
                =1                  POP     00H
                =1                  POP     ACC
                =1            ;
                =1   130      %*DEFINE(GET_AMU)
                =1                  MOVX    A,@DPTR              %'GET THE AMU HIGH BYTE%'
                =1                  MOV     AMU_H,A
                =1                  INC     DPTR
                =1                  MOVX    A,@DPTR              %'GET THE AMU LOW BYTE%'
                =1                  MOV     AMU_L,A
                =1                  INC     DPTR
                =1
                =1   131      %*DEFINE(GET_RAT_AND_BIASES)LOCAL GET_NEXT(
                =1                  MOV     R2,#05H
                =1                  MOV     R0,#RAT_H
                =1            %GET_NEXT:
                =1                  MOVX    A,@DPTR
                =1                  INC     DPTR
                =1                  MOV     @R0,A                %'SAVE THE PARAMETER READY%'
                =1                  INC     R0                   %'POINT TO NEXT PARAMETER%'
                =1                  DJNZ    R2,%GET_NEXT
                =1            ;
                =1   132      %*DEFINE(SET_OFFSETS)(
                =1                  MOVX    A,@DPTR
                =1                  MOV     P0,ACC
                =1                  MOV     P2,#BIAS_1_SELECT
                =1                  MOV     P2,#00
                =1                  INC     DPTR
                =1                  MOVX    A,@DPTR
                =1                  MOV     P0,ACC
                =1                  MOV     P2,#BIAS_1_SELECT+1
                =1                  MOV     P2,#00
                =1                  INC     DPTR
                =1                  MOVX    A,@DPTR
                =1                  MOV     P0,ACC
                =1                  MOV     P2,#BIAS_1_SELECT+2
                =1                  MOV     P2,#00
                =1                  INC     DPTR
                =1                  MOVX    A,@DPTR
                =1                  MOV     P0,ACC
```

```
LOC  OBJ            LINE     SOURCE

                   =1                 MOV     P2,#BIAS_1_SELECT+3
                   =1                 MOV     P2,#00
                   =1                 INC     DPTR
                   =1        ;
                   =1   133  %*DEFINE(SET_AMU_VR)(
                   =1        %'............... PUT THE AMU AND VR VALUES INTO THE DACS ...............%'
                   =1                 MOV     P0,AMU_H
                   =1                 MOV     P2,#MSB_RF
                   =1                 MOV     P2,#00
                   =1                 MOV     P0,AMU_L
                   =1                 MOV     P2,#LSB_RF
                   =1                 MOV     P2,#00
                   =1                 MOV     P0,RAT_H
                   =1                 MOV     P2,#MSB_DC
                   =1                 MOV     P2,#00
                   =1                 MOV     P0,RAT_L
                   =1                 MOV     P2,#LSB_DC
                   =1                 MOV     P2,#00
                   =1        )
                   =1   134  %*DEFINE(SET_BIASES)(
                   =1                 MOV     A,0_BIAS
                   =1                 MOV     C,ACC.7
                   =1                 MOV     ION_CONT,C      %' SELECT POSITIVE OR NEGATIVE ION MODE %'
                   =1                 MOV     P0,0_BIAS
                   =1                 MOV     P2,#BIAS_0_SELECT
                   =1                 MOV     P2,#00H
                   =1                 MOV     P0,BIAS_1
                   =1                 MOV     P2,#BIAS_0_SELECT+1
                   =1                 MOV     P2,#00H
                   =1                 MOV     P0,BIAS_2
                   =1                 MOV     P2,#BIAS_0_SELECT+2
                   =1                 MOV     P2,#00H
                   =1                 MOV     P0,BIAS_3
                   =1                 MOV     P2,#BIAS_0_SELECT+3
                   =1        )
                   =1   135  ;################################################################################
                        136  $INCLUDE( :F1:LIPD.BT)
----               =1   137  CSEG AT 00
                   =1   138  ;
                   =1   139  ;
0000 020035        =1   140           JMP     TO_BOOT
                   =1   141  ;
                   =1   142  ;
0003               =1   143  ORG     EXTI0
0003 020000      F =1   144           JMP     PCM_ROUTINE
```

```
LOC  OBJ            LINE     SOURCE

               =1    145     ;
0013           =1    146     ORG     EXTI1
0013 32        =1    147             RETI
               =1    148     ;
               =1    149     ;
               =1    150     ;
0023           =1    151     ORG     SINT
0023 020000  F =1    152             JMP     SERIAL
               =1    153     ;
               =1    154     ;
0035           =1    155     ORG 35H
               =1    156     ;
               =1    157     ;
               =1    158     TO_BOOT:
               =1    159     ;
               =1    160     USING 0
0035 C2D3      =1    161             CLR     RS0
0037 C2D4      =1    162             CLR     RS1
0039 75A000    =1    163             MOV     P2,#00H          ;CLEAR OUT RESET CLEAR
003C D296      =1    164             SETB    OD               ;PREVENT THE CAP BLOWING
003E C295      =1    165             CLR     STROBE           ;DONT STROBE ANYTHING INTO THE DRIVER
0040 D2B5      =1    166             SETB    RF_HV            ;TURN OFF THE VOLTAGES
0042 C292      =1    167             CLR     MON_OE
0044 C294      =1    168             CLR     MON_ALE
0046 C290      =1    169             CLR     XFR
0048 D2B8      =1    170             SETB    PX0              ;################################
004A D288      =1    171             SETB    IT0              ; READY INTO FOR WATCH DOG CIRCUIT
004C D2A8      =1    172             SETB    EX0
004E D2AF      =1    173             SETB    EA
0050 D289      =1    174             SETB    IE0              ;CLEAR OUT WATCH DOG WITH INT0 INTERUPT
0052 00        =1    175             NOP
0053 00        =1    176             NOP
0054 904000    =1    177             MOV     DPTR,#4000H      ;ENTER THE PCM SYNC WORDS
0057 E0        =1    178             MOVX    A,@DPTR
0058 F539      =1    179             MOV     QUEUE,A
005A A3        =1    180             INC     DPTR
005B E0        =1    181             MOVX    A,@DPTR
005C F53A      =1    182             MOV     QUEUE+1,A
005E 750839    =1    183             MOV     PCM_POINT,#QUEUE
0061 751100    =1    184             MOV     CAP_SAFETY,#00  ;PREVENT BLOWING OF CAP
0064 7580FF    =1    185             MOV     P0,#0FFH                   ;RESET PORTS IF AN ESCAPE CMD
0067 75A000    =1    186             MOV     P2,#00H
006A 758150    =1    187             MOV     SP,#STACK        ;RESET THE STACK POINTER
               =1    188     ;################################################################################
006D 020000  F =1    189             JMP     SERIAL_BOOT
```

```
LOC  OBJ            LINE     SOURCE

               =1    190     ;######################################################################
               =1    191     ; FOR TESTING PURPOSES
 0005          =1    192     TEST_TIME        EQU     5D
0070 7805      =1    193            MOV     R0,#TEST_TIME
0072 7905      =1    194            MOV     R1,#TEST_TIME
0074 7A05      =1    195            MOV     R2,#TEST_TIME
0076 D2B4      =1    196     IF_FLT: SETB   TST              ;TEST FOR #TEST TIME
0078 20B406    =1    197            JB      TST,DEC_RTN
007B 1A        =1    198            DEC     R2               ; DEC IF TEST
007C D9F8      =1    199            DJNZ    R1,IF_FLT        ; LOOP TO TEST PIN
007E 020084    =1    200            JMP     TO_BOOT_IT
               =1    201     ;
               =1    202     ;
               =1    203     DEC_RTN:
0081 18        =1    204            DEC     R0               ;DECREMENT IF FLIGHT
0082 D9F2      =1    205            DJNZ    R1,IF_FLT        ;LOOP TO TEST 5 TIMES
               =1    206     ;
               =1    207     ;
               =1    208     TO_BOOT_IT:                     ;IF EITHR TEST IS VALID FOR TEST TIME TIMES
               =1    209     ;                               THEN BOOT PROPER FOR TYPE
               =1    210     ;
0084 EA        =1    211            MOV     A,R2
0085 6000    F =1    212            JZ      SERIAL_BOOT      ;IF VALID 5 TIMES THEN BOOT FOR TEST
               =1    213     MAY_FLIGHT:
0087 E8        =1    214            MOV     A,R0             ;ELSE
0088 7003      =1    215            JNZ     TO_RESET_PAYLOAD
008A 020000  F =1    216            JMP     FLIGHT_BOOT      ;IF NEITHER IS VALID THEN RECYCLE
               =1    217     TO_RESET_PAYLOAD:
008D 0100      =1    218            JMP     RESET
               =1    219     ;
               =1    220     ;
                     221     $INCLUDE( :F1:SERIAL.BT)
               =1    222     ;
               =1    223     ;
               =1    224     ;
               =1    225     ;
               =1    226     ;###### A BENCH TEST IS IN PROGRESS ######
               =1    227     ;
----           =1    228     $SEG SERIAL_BOOT_CODE
               =1    229     ;
               =1    230     SERIAL_BOOT:
0000 D210      =1    231            SETB    CMD_RDY          ;PRESET SRIAL FLAGS
0002 C211      =1    232            CLR     XBF
0004 753000    =1    233            MOV     CMD_BUF,#00H
0007 753800    =1    234            MOV     CMD_CNT,#00H
```

```
MCS-51 MACRO ASSEMBLER      LIPD (SUPER ARCAS)                                    [illegible] OCT 85   PAGE [illegible]


LOC  OBJ            LINE     SOURCE

000A 754F00     =1   235              MOV     Q_PTR,#00H
000D 758920     =1   236              MOV     TMOD,#20H        ;TIMER 1 MODE 2
0010 758BF3     =1   237              MOV     TL1,#BAUD_COUNT  ;SET UP FOR 1200 BAUD
0013 758DF3     =1   238              MOV     TH1,#BAUD_COUNT
0016 759850     =1   239              MOV     SCON,#50H        ;8 BIT NO PARITY SERIAL LINK
0019 D28E       =1   240              SETB    TR1              ;TURN ON TIMER 1
001B C299       =1   241              CLR     TI               ;CLEAR OUT INTERUPT FLAGS
001D C298       =1   242              CLR     RI
001F C2BC       =1   243              CLR     PS               ;SERIAL PORT LOW PRIORITY
0021 751100     =1   244              MOV     CAP_SAFETY,#00H
0024 D2AC       =1   245              SETB    ES               ;ENABL SERIAL INTERUPT
                =1   246      FLIGHT_BOOT:
0026 20AC03     =1   247              JB      ES,KEEP_SAFETY  ;IF IN A BENCH TEST MODE KEEP CAP SAFETY=00H
0029 7511FF     =1   248              MOV     CAP_SAFETY,#0FFH
                =1   249      KEEP_SAFETY:
002C D2B8       =1   250              SETB    PX0              ;INT 0 HI PRIORITY
002E D288       =1   251              SETB    IT0              ;INTERUPT ON FALLING EDGE INT0
0030 750839     =1   252              MOV     PCM_POINT,#QUEUE;POINT THE PCM TO WORD 0
0033 D2A8       =1   253              SETB    EX0              ;ENABLE INT0
0035 D2AF       =1   254              SETB    EA               ;ENABLE INTERUPTS
0037 30AC05     =1   255              JNB     ES,FLT_DELAY     ;IF A FLIGHT THEN JMP TO DELAY
                =1   256      ;
                =1   257      ;
                =1   258      WAIT:
003A 00         =1   259              NOP
003B 00         =1   260              NOP
003C 00         =1   261              NOP
003D 80FB       =1   262              JMP     WAIT             ;IF IN BENCH TEST MODE WAIT FOR COMMAND
                =1   263      ;
                =1   264      ;
                =1   265      FLT_DELAY:
003F 904002     =1   266              MOV     DPTR,#4002H
0042 E0         =1   267              MOVX    A,@DPTR          ;GET THE FRAME SYNC WORD 0
0043 F539       =1   268              MOV     QUEUE,A
0045 A3         =1   269              INC     DPTR
0046 E0         =1   270              MOVX    A,@DPTR          ;GET FRAME WORD 1
0047 F53A       =1   271              MOV     QUEUE+1,A
0049 A3         =1   272              INC     DPTR
004A E0         =1   273              MOVX    A,@DPTR          ;SET THE DELAY TIME1 FROM EEPROM
                =1   274              ZWAIT(A)                 ;WAIT EEPROM 002 /10 SECONDS
006A A3         =1   296              INC     DPTR
006B E0         =1   297              MOVX    A,@DPTR
                =1   298              ZWAIT(A)                 ;WAIT EEPROM 003 /10 SECONDS
                =1   320      BLOW_CAP:
008B 74FF       =1   321              MOV     A,#0FFH
```

```
LOC  OBJ            LINE     SOURCE

00BD 5511        =1  322              ANL     A,CAP_SAFETY    ;SAFETY FEATURE
00BF F580        =1  323              MOV     P0,A
00C1 D295        =1  324              SETB    STROBE
00C3 C295        =1  325              CLR     STROBE          ;LATCH IN THE VALUE
00C5 C296        =1  326              CLR     OD              ;BLOW CAP ##############################
00C7 A3          =1  327              INC     DPTR
00C8 E0          =1  328              MOVX    A,@DPTR
                 =1  329              ZWAIT(A)                ;WAIT EEPROM 004 *10 SECONDS
                 =1  351     BLOW_CAP_AGAIN:
00E8 74FF        =1  352              MOV     A,#0FFH
00EA 5511        =1  353              ANL     A,CAP_SAFETY
00EC F580        =1  354              MOV     P0,A
00EE D295        =1  355              SETB    STROBE
00F0 C295        =1  356              CLR     STROBE          ;CAP IS BLOWN IF SAFETY 2LSB=1
00F2 A3          =1  357              INC     DPTR
00F3 E0          =1  358              MOVX    A,@DPTR
                 =1  359              ZWAIT(A)                ;WAIT EEPROM 005 *10 SECONDS
00E3 C2B5        =1  381              CLR     RF_HV           ;TURN ON THE RF AND HIGH VOLTAGE
00E5 020000   F  =1  382              JMP     DC_PROFILE
                     383     $INCLUDE(:F1:LIFD.SER
                 =1  384     ;
                 =1  385     ;
00E8 C0E0        =1  386     SERIAL: PUSH    ACC
00EA C0D0        =1  387              PUSH    PSW
                 =1  388     USING   0
00EC C2D3        =1  389              CLR     RS0
00EE C2D4        =1  390              CLR     RS1
00F0 20980A      =1  391              JB      RI,RECIEVE      ;IF DATA/CMD INCOMING THE JMP
00F3 C299        =1  392              CLR     TI
00F5 120000   F  =1  393              CALL    XMIT            ;ELSE TRNSMIT NEXT IN QUEUE
00F8 D0D0        =1  394              POP     PSW
00FA D0E0        =1  395              POP     ACC
00FC 32          =1  396              RETI                    ;AND GO BACK TO WAITING
                 =1  397     ;
                 =1  398     ;
                 =1  399     RECIEVE:
00FD C298        =1  400              CLR     RI
00FF 8599F0      =1  401              MOV     B,SBUF          ;SAVE INCOMING IN B
0102 20F7??      =1  402              JB      B.7,CMD         ;JUMP IF A COMMAND IS RECIEVED
0105 201005      =1  403              JB      CMD_RDY,RCV_ERR ;IF EXPECTING A COMMAND AND RECIEVE
                 =1  404                                      ;DATA THEN ERR
0108 E53B        =1  405              MOV     A,CMD_CNT
010A B40B03      =1  406              CJNE    A,#0BH,NO_ERR   ;IF TOO MANY PIECES OF DATA ERROR
                 =1  407     RCV_ERR:
010D 020000   F  =1  408              JMP     ERR             ;= 8
```

```
LOC  OBJ            LINE     SOURCE

0110 5005        =1   409    NO_ERR: JNC     TO_ERR          ;> 8
0112 E5F0        =1   410            MOV     A,B
0114 B44703      =1   411            CJNE    A,#6,LT_6
0117 020000    F =1   412    TO_ERR: JMP     ERR             ;IF DATA =6 THEN ERR
011A 4003        =1   413    LT_6:   JC      IS_LT_6         ;< 6
011C 020000    F =1   414            JMP     ERR             ;>=6
                 =1   415    IS_LT_6:
011F B44003      =1   416            CJNE    A,#AA-1H,NOT_LT_A
0122 020000    F =1   417            JMP     ERR             ;=A-1
                 =1   418    NOT_LT_A:
0125 5016        =1   419            JNC     VALID           ;A'= V  =F
0127 B43A03      =1   420            CJNE    A,#NIN+1,LT_9
012A 020000    F =1   421            JMP     ERR             ;9< V  A
012D 4003        =1   422    LT_9:   JC      IS_LT_9
012F 020000    F =1   423            JMP     ERR             ;9< V  A
                 =1   424    IS_LT_9:
0132 B42F03      =1   425            CJNE    A,#ZER-1H,GT_0
0135 020000    F =1   426            JMP     ERR
0138 5003        =1   427    GT_0:   JNC     VALID           ;0 = V  =A
013A 020000    F =1   428            JMP     ERR             ;0< V
                 =1   429    VALID:
013D E538        =1   430            MOV     A,CMD_CNT       ;GET THE CMD/DATA POINTER
013F 2431        =1   431            ADD     A,#CMD_BUF+1    ;MAKE IT POINT TO THE ADDRESS / DATA SECTION
0141 F8          =1   432            MOV     R0,A
0142 E5F0        =1   433            MOV     A,B
0144 F6          =1   434            MOV     @R0,A           ;AND SAVE THE DATA IN THE BUFFER
0145 0538        =1   435            INC     CMD_CNT         ;POINT TO THE NEXT POSITION
0147 120000    F =1   436            CALL    ECHO            ;ECHO BACK THE DATA
014A D0D0        =1   437            POP     PSW
014C D0E0        =1   438            POP     ACC
014E 32          =1   439            RETI                    ;RETURN FOR MORE DATA OR EOT CODE
                 =1   440    ;
                 =1   441    ;
                 =1   442    XMIT:
014F 7FFF        =1   443            MOV     R7,#0FFH        ;WAIT FOR APPROX 1MS
0151 DFFE        =1   444            DJNZ    R7,$            ;TO ALLOW SEPARATION OF XMITTED DATA
0153 E54F        =1   445            MOV     A,Q_PTR         ;CHECK TO SEE IF QUEUE IS EMPTY
0155 7003        =1   446            JNZ     Q_NOT_EMPTY
0157 C211        =1   447            CLR     XBF             ;NOTHING IN THE PROCESS OF BEING XMITTED
0159 22          =1   448            RET
                 =1   449    Q_NOT_EMPTY:
015A 853999      =1   450            MOV     SBUF,QUEUE      ;GET THE NEXT PIECE OF DATA IN LINE FOR XMIT
                 =1   451            ZPOP                    ;TRANSMIT THE NEXT (FIRST IN .. LAST OUT
017B D211        =1   473            SETB    XBF             ;SHOW THE XMIT AS BUSY
017D 22          =1   474            RET                     ;AND LEAVE
```

```
LOC  OBJ            LINE    SOURCE

                 =1  475    ;
                 =1  476    ;
                 =1  477    ;
                 =1  478    CMD:
017E E5F0        =1  479            MOV     A,B
0180 B47F03      =1  480            CJNE A,#ESC_CODE.NOT_ESC
0183 020000    F =1  481            JMP     ESC_CMD
                 =1  482    NOT_ESC:
0186 B4A810      =1  483            CJNE    A,#EOT_CODE,NOT_EOT
0189 C082        =1  484            PUSH    DPL
018B C083        =1  485            PUSH    DPH
018D 120000    F =1  486            CALL    EOT_CMD
0190 D083        =1  487            POP     DPH
0192 D082        =1  488            POP     DPL
0194 D0D0        =1  489            POP     PSW
0196 D0E0        =1  490            POP     ACC
0198 32          =1  491            RETI
                 =1  492    NOT_EOT:
0199 753B00      =1  493            MOV     CMD_CNT,#00H            ;RESET TO ACCEPT DATA FOR A NEW COMMAND
019C F530        =1  494            MOV     CMD_BUF,A               ;SAVE THE COMMAND IN THE 1ST POSITION
019E E53B        =1  495            MOV     A,CMD_CNT
01A0 C210        =1  496            CLR     CMD_RDY
01A2 120000    F =1  497            CALL    ECHO
01A5 D0D0        =1  498            POP     FSW
01A7 D0E0        =1  499            POP     ACC
01A9 32          =1  500            RETI
                 =1  501    ;
                 =1  502    ESC_CMD:
01AA 75F07F      =1  503            MOV     B,#ESC_CODE
01AD 120000    F =1  504            CALL    ECHO
01B0 758150      =1  505            MOV SP,#STACK
01B3 7400        =1  506            MOV     A,#00
01B5 C0E0        =1  507            PUSH    ACC
01B7 C0E0        =1  508            PUSH    ACC
01B9 900000    F =1  509            MOV     DPTR,#PRE_ESC
01BC C082        =1  510            PUSH    DPL
01BE C083        =1  511            PUSH    DPH
01C0 32          =1  512            RETI                    ;ABORT EITHER INTERUPT
                 =1  513    PRE_ESC:
01C1 2011FD      =1  514            JB      1BF,$           ;ESC ECHO SENT
01C4 C2AF        =1  515            CLR     EA
01C6 32          =1  516            RETI                    ;AND RETURN TO RESET
                 =1  517    ;
                 =1  518    ;
                 =1  519    ECHO:
```

```
LOC  OBJ            LINE     SOURCE

01C7 C0E0       =1   520              PUSH    ACC
01C9 C000       =1   521              PUSH    00H
01CB E54F       =1   522              MOV     A,Q_PTR
01CD B41503     =1   523              CJNE    A,#Q_PTR-QUEUE-1,ROOM_IN_Q
01D0 020000   F =1   524              JMP     Q_FULL
                =1   525     ROOM_IN_Q:
01D3 2439       =1   526              ADD     A,#QUEUE                ;POINT INTO THE QUEUE
01D5 F8         =1   527              MOV     R0,A
01D6 A6F0       =1   528              MOV     @R0,B                   ;AND ENTER THE ECHO
01D8 054F       =1   529              INC     Q_PTR
01DA E54F       =1   530              MOV     A,Q_PTR
01DC B40105     =1   531              CJNE    A,#01H,GT1              ;CREATE A XMIT INT IF FIRST ENTRY
01DF 201102     =1   532              JB      YBF,GT1                 ;ONLY CREATE AN INTERUPT IF IN PROCESS
                =1   533                                              ;OF XMISSION
01E2 D299       =1   534              SETB    TI                      ;CAUSE AN INTERUPT UPON RETURN
                =1   535     GT1:
01E4 D000       =1   536              POP     00H
01E6 D0E0       =1   537              POP     ACC
01E8 22         =1   538              RET
                =1   539     ;
01E9 3099FD     =1   540     Q_FULL: JNB      TI,Q_FULL
01EC 120000   F =1   541              CALL    XMIT
01EF D000       =1   542              POP     00H
01F1 D0E0       =1   543              POP     ACC
01F3 80D2       =1   544              JMP     ECHO
                =1   545     ;
                =1   546     ERR:
01F5 C0E0       =1   547              PUSH    ACC
01F7 E5F0       =1   548              MOV     A,B
01F9 B47F02     =1   549              CJNE    A,#ESC_CODE,NO_ESC_ERR
01FC 80AC       =1   550              JMP     ESC_CMD
                =1   551     NO_ESC_ERR:
01FE D0E0       =1   552              POP     ACC
0200 75F0FF     =1   553              MOV     B,#ERR_CODE
0203 120000   F =1   554              CALL    ECHO
0206 753800     =1   555              MOV     CMD_CNT,#00H    ;INVALIDATE CURRENT COMMAND
0209 753000     =1   556              MOV     CMD_BUF,#00H    ;POINT TO THE FIRST DATA POSITION
020C D210       =1   557              SETB    CMD_RDY         ;ACCEPT ONLY A COMMAND AS NEXT
                =1   558     FORCE_RETURN:
020E 758154     =1   559              MOV     SP,#STACK+4
0211 D0D0       =1   560              POP     PSW
0213 D0E0       =1   561              POP     ACC
0215 32         =1   562              RETI
                =1   563     ;
                =1   564     ;
```

```
LOC  OBJ            LINE    SOURCE

                =1   565    EOT_CMD:
0216 E530       =1   566            MOV     A,CMD_BUF         ;GET THE CMD'S VALUE
0218 753000     =1   567            MOV     CMD_BUF,#00H      ;REPLACE WITH A ZERO FLAG
021B 7002       =1   568            JNZ     CMD_SET           ;IF NO CMD THEN ERR
021D 80D6       =1   569            JMP     ERR               ;IF CMD=0 THE NO CMD SO ERR
                =1   570    CMD_SET:
  000E          =1   571    NO_OF_CMDS      EQU     14D       ;15 COMMANDS 0-14 ARE VALID
021F C2E7       =1   572            CLR     ACC.7             ;STRIP THE MSB
0221 23         =1   573            RL      A
0222 23         =1   574            RL      A                 ;PROVIDE A X4 OFFSET
0223 900000   F =1   575            MOV     DPTR,#TOP_OF_JMP_TBLE
0226 73         =1   576            JMP     @A+DPTR           ;JUMP TO PROPER VECTOR
                =1   577    TOP_OF_JMP_TBLE:
0227 020000   F =1   578            JMP     CMD0
022A 00         =1   579            NOP
022B 020000   F =1   580            JMP     CMD1
022E 00         =1   581            NOP
022F 020000   F =1   582            JMP     CMD2
0232 00         =1   583            NOP
0233 020000   F =1   584            JMP     CMD3
0236 00         =1   585            NOP
0237 020000   F =1   586            JMP     CMD4
023A 00         =1   587            NOP
023B 020000   F =1   588            JMP     CMD5
023E 00         =1   589            NOP
023F 020000   F =1   590            JMP     CMD6
0242 00         =1   591            NOP
0243 020000   F =1   592            JMP     CMD7
0246 00         =1   593            NOP
0247 020000   F =1   594            JMP     CMD8
024A 00         =1   595            NOP
024B 020000   F =1   596            JMP     CMD9
024E 00         =1   597            NOP
024F 020000   F =1   598            JMP     CMD10
0252 00         =1   599            NOP
0253 020000   F =1   600            JMP     CMD11
0256 00         =1   601            NOP
0257 020000   F =1   602            JMP     CMD12
025A 00         =1   603            NOP
025B 020000   F =1   604            JMP     CMD13
025E 00         =1   605            NOP
025F 020000   F =1   606            JMP     CMD14
0262 00         =1   607            NOP
0263 00         =1   608            NOP
0264 00         =1   609            NOP
```

```
LOC  OBJ           LINE     SOURCE

0265 00          =1  610            NOP
0266 00          =1  611            NOP
0267 00          =1  612            NOP
0268 00          =1  613            NOP
0269 00          =1  614            NOP
026A 00          =1  615            NOP
026B 8088        =1  616            JMP     ERR
                 =1  617    ;
                 =1  618    ;
                 =1  619    ;
                 =1  620    ;
026D C212        =1  621    CMD0:   CLR     CMD1FLG         ;POKE COMMAND
                 =1  622    CMD_1_DECODE:
026F E531        =1  623            MOV     A,CMD_BUF+1     ;POKE COMMAND... START DECODING ADDRESS
0271 120000    F =1  624            CALL    DECODE          ;CHANGE ASCII CHARACTER INTO HEX VALUE
0274 5407        =1  625            ANL     A,#07H          ;STRIP NOT APPLICABLE BITS
0276 F531        =1  626            MOV     CMD_BUF+1,A
0278 E532        =1  627            MOV     A,CMD_BUF+2
027A 120000    F =1  628            CALL    DECODE
027D C4          =1  629            SWAP    A
027E F532        =1  630            MOV     CMD_BUF+2,A
0280 E533        =1  631            MOV     A,CMD_BUF+3
0282 120000    F =1  632            CALL    DECODE
0285 4232        =1  633            ORL     CMD_BUF+2,A
0287 201228      =1  634            JB      CMD1FLG,CMD1_DECODED    ;ADDRESS IS DECODED INTO CMD_BUF+1 & +2
028A E534        =1  635            MOV     A,CMD_BUF+4
028C 120000    F =1  636            CALL    DECODE
028F C4          =1  637            SWAP    A
0290 F533        =1  638            MOV     CMD_BUF+3,A
0292 E535        =1  639            MOV     A,CMD_BUF+5
0294 120000    F =1  640            CALL    DECODE
0297 4233        =1  641            ORL     CMD_BUF+3,A     ;DATA AND ADDRESS DECODED
0299 853183      =1  642            MOV     DPH,CMD_BUF+1
029C 853282      =1  643            MOV     DPL,CMD_BUF+2
029F 438340      =1  644            ORL     DPH,#EEPROM_SELECT
02A2 E533        =1  645            MOV     A,CMD_BUF+3
02A4 F0          =1  646            MOVX    @DPTR,A         ;AND WRITE IT TO THE EEPROM
                 =1  647    EEPROM_RET:
02A5 75F0AB      =1  648            MOV     B,#EOT_CODE
02A8 120000    F =1  649            CALL    ECHO            ;ECHO THE EOT CODE TO SHOW CMD COMPLETE
02AB D210        =1  650            SETB    CMD_RDY
02AD 22          =1  651            RET                     ;AND LEAVE
                 =1  652    ;
                 =1  653    ;
02AE D212        =1  654    CMD1:   SETB    CMD1FLG         ;PEEK COMMAND
```

```
LOC  OBJ            LINE     SOURCE

02B0 80BD       =1   655            JMP     CMD_1_DECODE    ;DECODE THE SAME AS CMD0
                =1   656     CMD1_DECODED:
02B2 E531       =1   657            MOV     A,CMD_BUF+1
02B4 4440       =1   658            ORL     A,#EEPROM_SELECT
02B6 F583       =1   659            MOV     DPH,A
02B8 853282     =1   660            MOV     DPL,CMD_BUF+2
02BB E0         =1   661            MOVX    A,@DPTR
02BC F5F0       =1   662            MOV     B,A
02BE 120000   F =1   663            CALL    ECHO
02C1 75F0AB     =1   664            MOV     B,#EOT_CODE
02C4 120000   F =1   665            CALL    ECHO
02C7 22         =1   666            RET
                =1   667     ;
                =1   668     ;
                =1   669     CMD2:                          ; ADAC TEST ROUTINE
02C8 E531       =1   670            MOV     A,CMD_BUF+1     ;GET THE HIGH NIBBLE OF THE MONITOR #
02CA 120000   F =1   671            CALL    DECODE
02CD C4         =1   672            SWAP    A
02CE F531       =1   673            MOV     CMD_BUF+1,A
02D0 E532       =1   674            MOV     A,CMD_BUF+2
02D2 120000   F =1   675            CALL    DECODE
02D5 4531       =1   676            ORL     A,CMD_BUF+1
02D7 120000   F =1   677            CALL    GET_MON
02DA F5F0       =1   678            MOV     B,A
02DC 120000   F =1   679            CALL    ECHO
02DF 75F0AB     =1   680            MOV     B,#EOT_CODE
02E2 120000   F =1   681            CALL    ECHO
02E5 22         =1   682            RET
                =1   683     ;
                =1   684     ;
                =1   685     GET_MON:
02E6 4400       =1   686            ORL     A,#MON_SELECT
02E8 7580FF     =1   687            MOV     P0,#0FFH        ;MAKE SURE THE PORT IS CLEAR
02EB F5A0       =1   688            MOV     P2,A
02ED D294       =1   689            SETB    MON_ALE
02EF C294       =1   690            CLR     MON_ALE
02F1 D293       =1   691            SETB    MON_STRT
02F3 C293       =1   692            CLR     MON_STRT
02F5 75A000     =1   693            MOV     P2,#00H         ;DESELECT EVERYTHING
02F8 7FFF       =1   694            MOV     R7,#0FFH                ;PREPARE TO WAIT FOR 250 USEC
02FA DFFE       =1   695            DJNZ    R7,$
02FC 75A000     =1   696            MOV     P2,#MON_SELECT  ;RESELECT ADAC
02FF 7580FF     =1   697            MOV     P0,#0FFH        ;MAKE SURE P0 IS CLEAR
0302 D292       =1   698            SETB    MON_OE          ;ENABLE MONITOR OUTPUT
0304 E580       =1   699            MOV     A,P0            ;AND GET THE VALUE
```

```
LOC  OBJ            LINE     SOURCE

0306 C292        =1   700            CLR     MON_OE
0308 75A000      =1   701            MOV     P2,#00H
030B 22          =1   702            RET
                 =1   703     ;
                 =1   704     DECODE:
030C C3          =1   705            CLR     C
030D C0E0        =1   706            PUSH    ACC
030F 9441        =1   707            SUBB    A,#AA
0311 4005        =1   708            JC      NOT_A_F
0313 240A        =1   709            ADD     A,#0AH              ;IF A-F CORRECT FOR OFFSET
0315 1581        =1   710            DEC     SP                  ;RE-ALIGN THE STACK
0317 22          =1   711            RET
                 =1   712     NOT_A_F:
0318 D0E0        =1   713            POP     ACC
031A C3          =1   714            CLR     C
031B 9430        =1   715            SUBB    A,#ZER
031D 22          =1   716            RET
                 =1   717     ;
                 =1   718     ;
                 =1   719     CMD3:
031E 120000    F =1   720            CALL R12_BIT_DECODE
0321 C080        =1   721            PUSH    P0
0323 853180      =1   722            MOV     P0,CMD_BUF+1
0326 C0A0        =1   723            PUSH    P2
0328 75A0D8      =1   724            MOV     P2,#MSB_RF
032B D0A0        =1   725            POP     P2
032D 853280      =1   726            MOV     P0,CMD_BUF+2
0330 C0A0        =1   727            PUSH    P2
0332 75A058      =1   728            MOV     P2,#LSB_RF
0335 D0A0        =1   729            POP     P2
0337 D080        =1   730            POP     P0
0339 75F0A8      =1   731            MOV     B,#EOT_CODE
033C 120000    F =1   732            CALL    ECHO
033F C290        =1   733            CLR     XFR
0341 D290        =1   734            SETB    XFR
0343 22          =1   735            RET
                 =1   736     CMD4:
0344 120000    F =1   737            CALL R12_BIT_DECODE
0347 C080        =1   738            PUSH    P0
0349 853180      =1   739            MOV     P0,CMD_BUF+1
034C C0A0        =1   740            PUSH    P2
034E 75A0E0      =1   741            MOV     P2,#MSB_DC
0351 D0A0        =1   742            POP     P2
0353 853280      =1   743            MOV     P0,CMD_BUF+2
0356 C0A0        =1   744            PUSH    P2
```

```
LOC  OBJ            LINE     SOURCE

0358 75A060     =1   745            MOV     P2,#LSB_DC
035B D0A0       =1   746            POP     P2
035D D080       =1   747            POP     P0
035F 75F0A8     =1   748            MOV     B,#EOT_CODE
0362 120000   F =1   749            CALL    ECHO
0365 C290       =1   750            CLR     XFR
0367 D290       =1   751            SETB    XFR
0369 22         =1   752            RET
                =1   753     ;
                =1   754     ;
                =1   755     R12_BIT_DECODE:
036A E531       =1   756            MOV     A,CMD_BUF+1
036C 120000   F =1   757            CALL    DECODE
036F C4         =1   758            SWAP    A
0370 F531       =1   759            MOV     CMD_BUF+1,A
0372 E532       =1   760            MOV     A,CMD_BUF+2
0374 120000   F =1   761            CALL    DECODE
0377 4231       =1   762            ORL     CMD_BUF+1,A
0379 E533       =1   763            MOV     A,CMD_BUF+3
037B 120000   F =1   764            CALL    DECODE
037E C4         =1   765            SWAP    A
037F F532       =1   766            MOV     CMD_BUF+2,A
0381 22         =1   767            RET
                =1   768     ;
                =1   769     ;
                =1   770     CMD5:                                       ;OFFSET/BIAS POKE
0382 E531       =1   771            MOV     A,CMD_BUF+1
0384 120000   F =1   772            CALL    DECODE
0387 4468       =1   773            ORL     A,#BIAS_0_SELECT
0389 F531       =1   774            MOV     CMD_BUF+1,A
038B E532       =1   775            MOV     A,CMD_BUF+2
038D 120000   F =1   776            CALL    DECODE
0390 C4         =1   777            SWAP    A
0391 F532       =1   778            MOV     CMD_BUF+2,A
0393 E533       =1   779            MOV     A,CMD_BUF+3
0395 120000   F =1   780            CALL    DECODE
0398 4232       =1   781            ORL     CMD_BUF+2,A
039A C080       =1   782            PUSH    P0
039C 853280     =1   783            MOV     P0,CMD_BUF+2
039F C0A0       =1   784            PUSH    P2
03A1 8531A0     =1   785            MOV     P2,CMD_BUF+1
03A4 D0A0       =1   786            POP     P2
03A6 D080       =1   787            POP     P0
03A8 75F0A8     =1   788            MOV     B,#EOT_CODE
03AB 120000   F =1   789            CALL    ECHO
```

```
LOC  OBJ            LINE     SOURCE

03AE 22          =1  790            RET
                 =1  791     ;
                 =1  792     ;
                 =1  793     CMD6:                          ;BIAS/OFFSET POKE
03AF E531        =1  794            MOV     A,CMD_BUF+1
03B1 120000    F =1  795            CALL    DECODE
03B4 4470        =1  796            ORL     A,#BIAS_1_SELECT
03B6 F531        =1  797            MOV     CMD_BUF+1,A
03B8 E532        =1  798            MOV     A,CMD_BUF+2
03BA 120000    F =1  799            CALL    DECODE
03BD C4          =1  800            SWAP    A
03BE F532        =1  801            MOV     CMD_BUF+2,A
03C0 E533        =1  802            MOV     A,CMD_BUF+3
03C2 120000    F =1  803            CALL    DECODE
03C5 4232        =1  804            ORL     CMD_BUF+2,A
03C7 C080        =1  805            PUSH    P0
03C9 853280      =1  806            MOV     P0,CMD_BUF+2
03CC C0A0        =1  807            PUSH    P2
03CE 8531A0      =1  808            MOV     P2,CMD_BUF+1
03D1 D0A0        =1  809            POP     P2
03D3 D080        =1  810            POP     P0
03D5 75F0A8      =1  811            MOV     B,#EOT_CODE
03D8 120000    F =1  812            CALL    ECHO
03DB 22          =1  813            RET
                 =1  814     CMD7:                          ;RAM PEEK
03DC E531        =1  815            MOV     A,CMD_BUF+1
03DE 120000    F =1  816            CALL    DECODE
03E1 C4          =1  817            SWAP    A
03E2 F531        =1  818            MOV     CMD_BUF+1,A
03E4 E532        =1  819            MOV     A,CMD_BUF+2
03E6 120000    F =1  820            CALL    DECODE
03E9 4231        =1  821            ORL     CMD_BUF+1,A
03EB C000        =1  822            PUSH    00H                 ;R0 RB0
03ED A831        =1  823            MOV     R0,CMD_BUF+1
03EF E6          =1  824            MOV     A,@R0
03F0 D000        =1  825            POP     00                  ;R0 RB0
03F2 F5F0        =1  826            MOV     B,A
03F4 120000    F =1  827            CALL    ECHO
03F7 75F0A8      =1  828            MOV     B,#EOT_CODE
03FA 120000    F =1  829            CALL    ECHO
03FD 22          =1  830            RET
                 =1  831     ;
                 =1  832     ;
                 =1  833     CMD8:                          ;RAM POKE
03FE E531        =1  834            MOV     A,CMD_BUF+1
```

```
LOC  OBJ            LINE     SOURCE

0400 120000   F =1   835              CALL    DECODE
0403 C4         =1   836              SWAP    A
0404 F531       =1   837              MOV     CMD_BUF+1,A
0406 E532       =1   838              MOV     A,CMD_BUF+2
0408 120000   F =1   839              CALL    DECODE
040B 4231       =1   840              ORL     CMD_BUF+1,A
040D E533       =1   841              MOV     A,CMD_BUF+3
040F 120000   F =1   842              CALL    DECODE
0412 C4         =1   843              SWAP    A
0413 F532       =1   844              MOV     CMD_BUF+2,A
0415 E534       =1   845              MOV     A,CMD_BUF+4
0417 120000   F =1   846              CALL    DECODE
041A 4532       =1   847              ORL     A,CMD_BUF+2
041C C000       =1   848              PUSH    00                  ;R0 RB0
041E A831       =1   849              MOV     R0,CMD_BUF+1
0420 F6         =1   850              MOV     @R0,A
0421 D000       =1   851              POP     00                  ;R0 RB0
0423 75F0A8     =1   852              MOV     B,#EOT_CODE
0426 120000   F =1   853              CALL    ECHO
0429 22         =1   854              RET
                =1   855     ;
                =1   856     ;
                =1   857     ;
                =1   858     CMD9:                                ;ENABLE HVRF
042A C2B5       =1   859              CLR     RF_HV
042C 75F0A8     =1   860              MOV     B,#EOT_CODE
042F 120000   F =1   861              CALL    ECHO
0432 22         =1   862              RET
                =1   863     ;
                =1   864     ;
                =1   865     ;
                =1   866     CMD10:                               ;DISABLE HVRF
0433 D2B5       =1   867              SETB    RF_HV
0435 75F0A8     =1   868              MOV     B,#EOT_CODE
0438 120000   F =1   869              CALL    ECHO
043B 22         =1   870              RET
                =1   871     ;
                =1   872     ;
                =1   873     ;
                =1   874     CMD11:                               ;CLR POS NEG ION SELECT
043C C297       =1   875              CLR     ION_CONT
043E 75F0A8     =1   876              MOV     B,#EOT_CODE
0441 120000   F =1   877              CALL    ECHO
0444 22         =1   878              RET
                =1   879     ;
```

```
LOC  OBJ            LINE    SOURCE

               =1   880    ;
               =1   881    ;
               =1   882    CMD12:                          ;SET POS NEG ION SELECT
0445 D297      =1   883            SETB    ION_CONT
0447 75F0A8    =1   884            MOV     B,#EOT_CODE
044A 120000  F =1   885            CALL    ECHO
044D 22        =1   886            RET
               =1   887    ;
               =1   888    ;
               =1   889    ;
               =1   890    CMD13:                          ;STEP SERIAL COMMAND
044E E531      =1   891            MOV     A,CMD_BUF+1
0450 120000  F =1   892            CALL    DECODE
0453 F583      =1   893            MOV     DPH,A
0455 E532      =1   894            MOV     A,CMD_BUF+2     ;HI NIBBLE DPL
0457 120000  F =1   895            CALL    DECODE
045A C4        =1   896            SWAP    A
045B F582      =1   897            MOV     DPL,A
045D E533      =1   898            MOV     A,CMD_BUF+3
045F 120000  F =1   899            CALL    DECODE
0462 4282      =1   900            ORL     DPL,A
0464 D214      =1   901            SETB    SERIAL_STEP
0466 7400    F =1   902            MOV     A,#LOW(CALL_PROFILE)
0468 C0E0      =1   903            PUSH    ACC
046A 7400    F =1   904            MOV     A,#HIGH(CALL_PROFILE)
046C C0E0      =1   905            PUSH    ACC
046E C215      =1   906            CLR     MULTI_AMU
0470 32        =1   907            RETI                    ;RETURN SO AS TO BE ABLE TO INTERUPT VIA THE
               =1   908    ;                                SERIAL LINK
               =1   909    CALL_PROFILE:
0471 2011FD    =1   910            JB      XBF,$           ;LOOP UNTIL QUEUE IS EMPTY
0474 120000  F =1   911            CALL    FLIGHT_PROFILE
               =1   912    NEXT_STEP:
0477 754F14    =1   913            MOV     Q_PTR,#20D
047A D299      =1   914            SETB    TI              ;CREATE A SERIAL INTERUPT
               =1   915    ;PROCESSING INTERUPTS TO SEND QUEUE
047C 00        =1   916            NOP
047D 2011FD    =1   917            JB      XBF,$           ;LOOP UNTIL ALL DATA IS SENT
0480 301508    =1   918            JNB     MULTI_AMU,STEP_EXIT
0483 2011FD    =1   919            JB      XBF,$
0486 120000  F =1   920            CALL    ENTRY
0489 80EC      =1   921            JMP     NEXT_STEP
               =1   922    STEP_EXIT:
048B 75F0A8    =1   923            MOV     B,#EOT_CODE
048E 120000  F =1   924            CALL    ECHO
```

```
LOC  OBJ           LINE     SOURCE

0491 C214      =1   925            CLR     SERIAL_STEP
0493 22        =1   926            RET
               =1   927     ;
               =1   928     ;
               =1   929     ;
               =1   930     CMD14:                           ;PCM RUN COMMAND
0494 900000  F =1   931            MOV     DPTR,#JMP_TO_PROFILE
0497 C082      =1   932            PUSH    DPL
0499 C083      =1   933            PUSH    DPH
049B 32        =1   934            RETI                     ;ALLOW SERIAL COMMANDS TO INTERUPT
               =1   935     JMP_TO_PROFILE:
049C 020000  F =1   936            JMP     DO_PROFILE       ;RELEASE CONTROL TO FLIGHT PROGRAM
                    937     $INCLUDE( :F1:LIPD.FLT)
----           =1   938     RSEG PROFILE_FLT_CODE
               =1   939     ;
               =1   940     ;
               =1   941     ;
               =1   942     DO_PROFILE:
0000 C216      =1   943            CLR     TIME_BIT
0002 904010    =1   944            MOV     DPTR,#4010H
               =1   945     FLIGHT_PROFILE:
0005 438340    =1   946            ORL     DPH,#EEPROM_SELECT
0008 C215      =1   947            CLR     MULTI_AMU
000A D290      =1   948            SETB    XFR
000C 75A060    =1   949            MOV     P2,#D_LOW              ;CLEAR THE COUNTER
000F 75A000    =1   950            MOV     P2,#00
0012 7580FF    =1   951            MOV     P0,#0FFH
               =1   952            %GET_AMU
001D 750838    =1   960            MOV     PCM_POINT,#QUEUE-1     ;ALIGN THE PCM TRAIN
0020 3016FD    =1   961            JNB     TIME_BIT,$             ;WAIT HERE TILL POINTING TO WORD 0
               =1   962     ;
               =1   963     ;
               =1   964     ;  THIS IS T=0 ** WORD 0 IS IN THE PROCESS OF BEING SENT **
               =1   965     ENTRY:
0023 301503    =1   966            JNB     MULTI_AMU,SET_NEW_VALS  ;IF MULTI AMU THEN LEAVE BIASES
0026 020000  F =1   967            JMP     WAIT_T_1
               =1   968     SET_NEW_VALS:
               =1   969            %GET_RAT_AND_BIASES
               =1   979            %SET_OFFSETS
               =1  1001            %SET_AMU_VR              ;SO SEND OUT THE AMU/RATIO VOLTAGES
               =1  1016            %SET_BIASES
00AA D290      =1  1032     WAIT_T_1:SETB   XFR             ;TRANSFER THE RF AND DC CONTROL LEVELS
00AC C290      =1  1033            CLR     XFR
00AE D290      =1  1034            SETB    XFR
00B0 D216      =1  1035            SETB    TIME_BIT         ;MAKE SURE THE FRAME IS ALIGNED
```

```
LOC  OBJ            LINE     SOURCE

00B2 2016FD     =1  1036             JB      TIME_BIT,$       ; SO WAIT HERE UNTIL THINGS SETTLE
                =1  1037     ;
                =1  1038     ;  WORD 0 IS SENT BY PCM
                =1  1039     ;
00B5 C083       =1  1040             PUSH    DPH              ;SAVE DATA POINTER
00B7 C082       =1  1041             PUSH    DPL              :
00B9 904000     =1  1042             MOV     DPTR,#4000H      ; SET FRAME WORD
00BC E0         =1  1043             MOVX    A,@DPTR          :                0
00BD F539       =1  1044             MOV     QUEUE,A          ; RESTORE IT
00BF A3         =1  1045             INC     DPTR             ;SET FRAME WORD
00C0 E0         =1  1046             MOVX    A,@DPTR          :               1
00C1 D082       =1  1047             POP     DPL
00C3 D083       =1  1048             POP     DPH              ; RESTORE THE DATA POINTER
                =1  1049     WAIT_T_2:
00C5 3016FD     =1  1050             JNB     TIME_BIT,$       ;WAIT TO SEND WORD 1
00C8 C291       =1  1051             CLR     CNT              ;START COUNTING DATA T=T+1ms
                =1  1052     ;
                =1  1053     ;  WORD 1 IS SENT BY PCM
                =1  1054     ;
00CA F53A       =1  1055             MOV     QUEUE+1,A        ;WORD 1 IS SENT SO RESTORE WORD 1
00CC 2016FD     =1  1056             JB      TIME_BIT,$               ;WAIT TO WRITE WORD 2
                =1  1057     ;
                =1  1058     ; WORD 2 IS SENT BY PCM
                =1  1059     ;
                =1  1060     ;
00CF 3016FD     =1  1061             JNB     TIME_BIT,$       ;WAIT TO WRITE WORD 3
                =1  1062     ;
                =1  1063     ; WORD 3 IS SENT BY PCM
                =1  1064     ;
                =1  1065     ;
00D2 2016FD     =1  1066             JB      TIME_BIT,$       ;WAIT TO WRITE WORD 4
                =1  1067     ;
                =1  1068     ; WORD 4 IS SENT BY PCM
                =1  1069     ;
00D5 85093B     =1  1070             MOV     QUEUE+2,AMU_H    ;RESTORE WORDS 2 AND 3
00D8 850A3C     =1  1071             MOV     QUEUE+3,AMU_L
                =1  1072     ;
00DB 3016FD     =1  1073             JNB     TIME_BIT,$       ;WAIT TO WRITE WORD 5
                =1  1074     ;
                =1  1075     ; WORD 5 IS SENT BY PCM
                =1  1076     ;
00DE 850B3D     =1  1077             MOV     QUEUE+4,RAT_H    ;RESTORE WORD 4
00E1 850C3E     =1  1078             MOV     QUEUE+5,RAT_L    ;RESTORE WORD 5
                =1  1079     ;
00E4 2016FD     =1  1080             JB      TIME_BIT,$       ;WAIT TO WRITE WORD 6
```

```
LOC  OBJ            LINE     SOURCE

               =1  1081     ;
               =1  1082     ; WORD 6 IS SENT BY PCM
               =1  1083             %GET_AMU                 ;GET THE NEXT VALUES FOR AMU
               =1  1091     ;
00EF 3016FD    =1  1092             JNB     TIME_BIT,$
               =1  1093     ;
               =1  1094     ; WORD 7 IS SENT BY PCM
               =1  1095     ;
00F2 7406      =1  1096             MOV     A,#RF_MON
00F4 120000  F =1  1097             CALL    ADC_MON          ;START ADAC CONVERSION OF RF MONITOR
               =1  1098     ;
00F7 2016FD    =1  1099             JB      TIME_BIT,$
               =1  1100     ;
               =1  1101     ; WORD 8 IS SENT BY PCM
               =1  1102     ;
00FA 120000  F =1  1103             CALL    ADAC             ;GET THE RF MONITOR VALUE
00FD F541      =1  1104             MOV     QUEUE+8,A        ;AND RESTORE WORD 8
00FF 7402      =1  1105             MOV     A,#HV_1_MON
0101 120000  F =1  1106             CALL    ADC_MON          ;START ADAC CONVERSION OF HV 1 MONITOR
               =1  1107     ;
0104 3016FD    =1  1108             JNB     TIME_BIT,$
               =1  1109     ;
               =1  1110     ; WORD 9 IS SENT BY PCM
               =1  1111     ;
0107 120000  F =1  1112             CALL    ADAC             ;GET THE HV 1 MONITOR VALUE
010A F542      =1  1113             MOV     QUEUE+9,A        ;AND RESTORE WORD 9
010C 7403      =1  1114             MOV     A,#HV_2_MON      ;
010E 120000  F =1  1115             CALL    ADC_MON          ;START CONVERSION OF HV 2 MONITOR
               =1  1116     ;
0111 2016FD    =1  1117             JB      TIME_BIT,$
               =1  1118     ;
               =1  1119     ; WORD 10 IS SENT BY PCM
               =1  1120     ;
0114 120000  F =1  1121             CALL    ADAC             ;GET THE HV 2 MONITOR VALUE
0117 F543      =1  1122             MOV     QUEUE+10,A       ;AND RESTORE WORD 10
0119 7400      =1  1123             MOV     A,#COMB_MON
011B 120000  F =1  1124             CALL    ADC_MON          ;START CONVERSION OF COMB MONITOR
               =1  1125     ;
011E 3016FD    =1  1126             JNB     TIME_BIT,$
               =1  1127     ;
               =1  1128     ; WORD 11 SENT BY PCM
               =1  1129     ;
0121 120000  F =1  1130             CALL    ADAC             ;GET THE COMB MONITOR VALUE
0124 F544      =1  1131             MOV     QUEUE+11,A       ;AND RESTORE WORD 11
0126 7404      =1  1132             MOV     A,#DC_MON
```

```
LOC  OBJ            LINE    SOURCE

0128 120000   F =1  1133           CALL    ADC_MON         ;START CONVERSION OF THE D.C. MONITOR
                =1  1134    ;
012B 2016FD     =1  1135           JB      TIME_BIT,$
                =1  1136    ;
                =1  1137    ; WORD 12 SENT BY PCM
                =1  1138    ;
012E 120000   F =1  1139           CALL    ADAC            ;GET THE D.C. MONITOR
0131 F545       =1  1140           MOV     QUEUE+12,A      ;RESTORE WORD 12
0133 7401       =1  1141           MOV     A,#VCC_MON
0135 120000   F =1  1142           CALL    ADC_MON         ;START THE CONVERSION OF THE +- 15 V MONITOR
                =1  1143    ;
0138 3016FD     =1  1144           JNB     TIME_BIT,$
                =1  1145    ;
                =1  1146    ; WORD 13 SENT BY PCM
                =1  1147    ;
013B 120000   F =1  1148           CALL    ADAC            ;GET THE +- 15 V MONITOR
013E F546       =1  1149           MOV     QUEUE+13,A      ;RESTORE WORD 13
0140 7405       =1  1150           MOV     A,#BAT_MON
0142 120000   F =1  1151           CALL    ADC_MON         ;START THE CONVERSION OF THE BATTERY MONITOR
                =1  1152    ;
0145 2016FD     =1  1153           JB      TIME_BIT,$
                =1  1154    ;
                =1  1155    ; WORD 14 SENT BY PCM
                =1  1156    ;
0148 120000   F =1  1157           CALL    ADAC            ;GET THE BATTERY MONITOR VALUE
014B F547       =1  1158           MOV     QUEUE+14,A      ;RESTORE WORD 14
014D 7407       =1  1159           MOV     A,#TEMP_MON
014F 120000   F =1  1160           CALL    ADC_MON         ;START THE CONVERSION OF THE TEMP MONITOR
                =1  1161    ;
0152 3016FD     =1  1162           JNB     TIME_BIT,$
                =1  1163    ;
                =1  1164    ; WORD 15 SENT BY PCM
                =1  1165    ;
0155 120000   F =1  1166           CALL    ADAC            ;GET THE TEMP MONITOR VALUE
0158 F548       =1  1167           MOV     QUEUE+15,A      ;RESTORE WORD 15
                =1  1168    ;
015A 2016FD     =1  1169           JB      TIME_BIT,$
                =1  1170    ;
                =1  1171    ; WORD 16 SENT BYE PCM
                =1  1172    ;
015D 850049     =1  1173           MOV     QUEUE+16,Q_BIAS          ;RESTORE WORD 16
0160 E509       =1  1174           MOV     A,AMU_H         ;TEST THE NEXT AMU FOR A CONTINUATION
                =1  1175                                   ; ARE ALREADY SET
0162 6034       =1  1176           JZ      NOT_MULTI_AMU
0164 B4FF08     =1  1177           CJNE    A,#0FFH,NOT_END_PROFILE ;IF FF THEN NEXT IS START
```

```
LOC  OBJ           LINE     SOURCE

0167 904010     =1  1178             MOV     DPTR,#4010H      ;POINT TO NEW BEGINING
016A C215       =1  1179             CLR     MULTI_AMU
016C 020000   F =1  1180             JMP     NOT_MULTI_AMU
                =1  1181     NOT_END_PROFILE:
016F D215       =1  1182             SETB    MULTI_AMU
                =1  1183             %SET_AMU_VR
0195 020000   F =1  1198             JMP     WORD_17_WAIT
                =1  1199     NOT_MULTI_AMU:
0198 A3         =1  1200             INC     DPTR
0199 A3         =1  1201             INC     DPTR
019A A3         =1  1202             INC     DPTR
019B A3         =1  1203             INC     DPTR
019C A3         =1  1204             INC     DPTR             ;ADJUST THE DATA POINTER TO POINT TO THE
                =1  1205                                      ; TOP OF A PAGE
019D 5382F0     =1  1206             ANL     DPL,#0F0H
01A0 538307     =1  1207             ANL     DPH,#07H
01A3 E583       =1  1208             MOV     A,DPH            ;CHECK FOR EEPROM OVERFLOW
01A5 7003       =1  1209             JNZ     NOT_EEPROM_RESET
01A7 758210     =1  1210             MOV     DPL,#10H         ; RESET TO POINT AT 010
                =1  1211     NOT_EEPROM_RESET:
01AA 438340     =1  1212             ORL     DPH,#EEPROM_SELECT
01AD C215       =1  1213             CLR     MULTI_AMU
                =1  1214             %SET_AMU
                =1  1222     ;
                =1  1223     WORD_17_WAIT:
01B7 3016FD     =1  1224             JNB     TIME_BIT,$
                =1  1225     ;
                =1  1226     ; WORD 17 SENT BY PCM
                =1  1227     ;
01BA 850E4A     =1  1228             MOV     QUEUE+17,BIAS_1          ;RESTORE WORD 17 AND SAVE FOR NEXT
                =1  1229     ;
01BD 2016FD     =1  1230             JB      TIME_BIT,$
                =1  1231     ; WORD 18 SENT BY PCM
01C0 850F4B     =1  1232             MOV     QUEUE+18,BIAS_2
                =1  1233     ;#### IN THE PROCESS OF SENDING WORD 19 ####
                =1  1234     ;####    REALIGN TO SEND WORD 0 NEXT    ####
01C3 750839     =1  1235             MOV     PCM_POINT,#QUEUE
01C6 3016FD     =1  1236             JNB     TIME_BIT,$
                =1  1237     ; WORD 19 SENT BY PCM ##### GET NEW DATA FOR QUEUE
01C9 D291       =1  1238             SETB    CNT
01CB 7580FF     =1  1239             MOV     P0,#0FFH
01CE 75A04B     =1  1240             MOV     P2,#D_HIGH
01D1 85803F     =1  1241             MOV     QUEUE+6,P0               ;GET THE HIGH BYTE
01D4 75A000     =1  1242             MOV     P2,#00
01D7 7580FF     =1  1243             MOV     P0,#0FFH
```

```
LOC  OBJ            LINE     SOURCE

01DA 75A060     =1  1244            MOV     P2,#D_LOW               ;GET THE LOW BYTE AND CLEAR THE COUNTER
01DD 858040     =1  1245            MOV     QUEUE+7,P0
01E0 75A000     =1  1246            MOV     P2,#00
01E3 85104C     =1  1247            MOV     QUEUE+19,BIAS_3         ;RESTORE WORD 19
01E6 201403     =1  1248            JB      SERIAL_STEP,PROFILE_RETURN;IF CALLED THEN EXECUTE A RETURN
01E9 020000   F =1  1249            JMP     ENTRY                   ;EXECUTE NEXT PROFILE
                =1  1250     PROFILE_RETURN:
01EC 22         =1  1251            RET                             ;RETURN TO CALLING PROGRAM
                =1  1252     ;
                =1  1253     ADC_MON:                               ;START ADAC CONVERSION
01ED C294       =1  1254            CLR     MON_ALE
01EF 4400       =1  1255            ORL     A,#MON_SELECT
01F1 C0A0       =1  1256            PUSH    P2
01F3 F5A0       =1  1257            MOV     P2,A
01F5 D294       =1  1258            SETB    MON_ALE
01F7 C294       =1  1259            CLR     MON_ALE
01F9 D0A0       =1  1260            POP     P2
01FB D293       =1  1261            SETB    MON_STRT                        ;START THE CONVERSION
01FD C293       =1  1262            CLR     MON_STRT
01FF 22         =1  1263            RET
                =1  1264     ;
                =1  1265     ;
                =1  1266     ADAC:                                  ;GET CONVERTED ANALOG VALUE
0200 C0A0       =1  1267            PUSH    P2
0202 C080       =1  1268            PUSH    P0
0204 75A000     =1  1269            MOV     P2,#00
0207 7580FF     =1  1270            MOV     P0,#0FFH
020A D292       =1  1271            SETB    MON_OE
020C E580       =1  1272            MOV     A,P0
020E C292       =1  1273            CLR     MON_OE
0210 D080       =1  1274            POP     P0
0212 D0A0       =1  1275            POP     P2
0214 22         =1  1276            RET
                =1  1277     ;
                =1  1278     ;
                =1  1279     ; INTO DRIVEN ROUTINE FOR PCM LINK
                =1  1280     PCM_ROUTINE:
0215 C0D0       =1  1281            PUSH    PSW
                =1  1282     USING 1                                ;
0217 C2D4       =1  1283            CLR     RS1
0219 D2D3       =1  1284            SETB    RS0
021B C090       =1  1285            PUSH    P1
021D C292       =1  1286            CLR     MON_OE
021F C0E0       =1  1287            PUSH    ACC
0221 C0A0       =1  1288            PUSH    P2
```

```
LOC  OBJ            LINE     SOURCE

0223 C080       =1  1289            PUSH    P0
0225 E6         =1  1290            MOV     A,@R0                   ;GET THE NEXT WORD
0226 08         =1  1291            INC     R0                      ;POINT TO THE NEXT WORD
0227 B84D02     =1  1292            CJNE    R0,#QUEUE+20,NOT_OVERFLOW
022A 7839       =1  1293            MOV     R0,#QUEUE               ;REALIGN IF OVERFLOW
                =1  1294     NOT_OVERFLOW:
022C 75A000     =1  1295            MOV     P2,#00
022F F580       =1  1296            MOV     P0,A
0231 75A078     =1  1297            MOV     P2,#PCM_LOAD            ;PUT THE WORD INTO THE PCM STREAM
0234 75A000     =1  1298            MOV     P2,#00
0237 B216       =1  1299            CPL     TIME_BIT                ;SHOW THAT THE WORD IS BEING SENT
0239 D080       =1  1300            POP     P0                      ;RESTORE THE WAY FOUND
023B D0A0       =1  1301            POP     P2
023D D0E0       =1  1302            POP     ACC
023F D090       =1  1303            POP     P1
0241 D0D0       =1  1304            POP     PSW
0243 32         =1  1305            RETI                            ;AND RETURN
                    1306     END
```

```
XREF SYMBOL TABLE LISTING
---- ------ ----- -------


N A M E           T Y P E  V A L U E       ATTRIBUTES AND REFERENCES

AA . . . . . . . .   NUMB    0041H   A        86# 416 707
ACC. . . . . . . . D ADDR    00E0H   A        276 294 300 318 331 349 361 379 386 395 438 490 499 507 508 520 537 543
                                              547 552 561 572 706 713 903 905 981 986 991 996 1019 1297 1302
ADAC . . . . . . . C ADDR    0200H   R       SEG=PROFILE_FLT_CODE  1103 1112 1121 1130 1139 1148 1157 1166 1266#
ADC_MON. . . . . . C ADDR    01EDH   R       SEG=PROFILE_FLT_CODE  1097 1106 1115 1124 1133 1142 1151 1160 1257#
AMU_H. . . . . . . D ADDR    0009H   A        20# 954 1003 1070 1085 1174 1185 1216
AMU_L. . . . . . . D ADDR    000AH   A        21# 957 1006 1071 1088 1188 1219
B. . . . . . . . . D ADDR    00F0H   A        401 402 410 433 479 503 528 548 553 648 662 664 678 681 731 748 789 811
                                              826 828 852 860 868 876 884 923
BAT_MON. . . . .     NUMB    0005H   A        96# 1150
BAUD_COUNT . . .     NUMB    00F2H   A        98# 237 238
BIAS_0_SELECT. .     NUMB    0068H   A        75# 773 1021 1024 1027 1030
BIAS_1_SELECT. .     NUMB    0070H   A        76# 796 982 987 992 997
BIAS_1 . . . . . . D ADDR    000EH   A        25# 1023 1228
BIAS_2 . . . . . . D ADDR    000FH   A        26# 1026 1232
BIAS_3 . . . . . . D ADDR    0010H   A        27# 1029 1247
BLOW_CAP_AGAIN . . C ADDR    00B6H   R       SEG=SERIAL_BOOT_CODE  351#
BLOW_CAP . . . . . C ADDR    008BH   R       SEG=SERIAL_BOOT_CODE  320#
CALL_PROFILE . . . C ADDR    0471H   R       SEG=SERIAL_BOOT_CODE  902 904 909#
CAP_DELAY. . . . . X ADDR    0FA2H   A        113#
CAP_SAFETY . . . . D ADDR    0011H   A        28# 184 244 248 322 353
CMD_1_DECODE . . . C ADDR    026FH   R       SEG=SERIAL_BOOT_CODE  622# 655
CMD_BUF. . . . . . D ADDR    0030H   A        33# 233 431 494 556 566 567 623 626 627 630 631 633 635 638 639 641 642
                                              643 645 657 660 670 673 674 676 722 726 739 743 756 759 760 762 763 766
                                              771 774 775 778 779 781 783 785 794 797 798 801 802 804 806 808 815 818
                                              819 821 823 834 837 838 840 841 844 845 847 849 891 894 898
CMD_CNT. . . . . . D ADDR    0038H   A        34# 234 405 430 435 493 495 555
CMD_RDY. . . . . . B ADDR    0022H.0 A        42# 231 403 496 557 650
CMD_SET. . . . . . C ADDR    021FH   R       SEG=SERIAL_BOOT_CODE  568 570#
CMD. . . . . . . . C ADDR    017EH   R       SEG=SERIAL_BOOT_CODE  402 478#
CMD0 . . . . . . . C ADDR    026DH   R       SEG=SERIAL_BOOT_CODE  578 621#
CMD1_DECODED . . . C ADDR    02B2H   R       SEG=SERIAL_BOOT_CODE  634 656#
CMD1 . . . . . . . C ADDR    02AEH   R       SEG=SERIAL_BOOT_CODE  580 654#
CMD10. . . . . . . C ADDR    0433H   R       SEG=SERIAL_BOOT_CODE  598 866#
CMD11. . . . . . . C ADDR    043CH   R       SEG=SERIAL_BOOT_CODE  600 874#
CMD12. . . . . . . C ADDR    0445H   R       SEG=SERIAL_BOOT_CODE  602 882#
CMD13. . . . . . . C ADDR    044EH   R       SEG=SERIAL_BOOT_CODE  604 890#
CMD14. . . . . . . C ADDR    0494H   R       SEG=SERIAL_BOOT_CODE  606 930#
CMD1FLG. . . . . . B ADDR    0022H.2 A        44# 621 634 654
CMD2 . . . . . . . C ADDR    02C8H   R       SEG=SERIAL_BOOT_CODE  582 669#
CMD3 . . . . . . . C ADDR    031EH   R       SEG=SERIAL_BOOT_CODE  584 719#
```

```
N A M E         T Y P E  V A L U E     ATTRIBUTES AND REFERENCES

CMD4 . . . . . . . C ADDR   0344H   R      SEG=SERIAL_BOOT_CODE  586 736#
CMD5 . . . . . . . C ADDR   0382H   R      SEG=SERIAL_BOOT_CODE  588 770#
CMD6 . . . . . . . C ADDR   03AFH   R      SEG=SERIAL_BOOT_CODE  590 793#
CMD7 . . . . . . . C ADDR   03DCH   R      SEG=SERIAL_BOOT_CODE  592 814#
CMD8 . . . . . . . C ADDR   03FEH   R      SEG=SERIAL_BOOT_CODE  594 833#
CMD9 . . . . . . . C ADDR   042AH   R      SEG=SERIAL_BOOT_CODE  596 858#
CNT. . . . . . . . B ADDR   0090H.1 A        57# 1051 1238
COMB_MON . . . .     NUMB   0000H   A        93# 1123
D_HIGH . . . . .     NUMB   0048H   A        69# 1240
D_LOW. . . . . .     NUMB   0060H   A        70# 949 1244
DC_MON . . . . .     NUMB   0004H   A        94# 1132
DEC_RTN. . . . . C ADDR   0081H   A        197 203#
DECODE . . . . . C ADDR   030CH   R      SEG=SERIAL_BOOT_CODE  624 628 632 636 640 671 675 704# 757 761 764 772 776
                                           780 795 799 803 816 820 835 839 842 846 892 895 899
DELAY_2. . . . . X ADDR   0FA3H   A        114#
DELAY_3. . . . . X ADDR   0FA4H   A        116#
DO_PROFILE . . . C ADDR   0000H   R      SEG=PROFILE_FLT_CODE  382 936 942#
DPH. . . . . . . . D ADDR   0083H   A        485 487 511 642 644 659 893 933 946 1040 1048 1207 1208 1212
DPL. . . . . . . . D ADDR   0082H   A        484 488 510 643 660 897 900 932 1041 1047 1206 1210
EA . . . . . . . . B ADDR   00A8H.7 A        173 254 515
ECHO . . . . . . . C ADDR   01C7H   R      SEG=SERIAL_BOOT_CODE  436 497 504 519# 544 554 649 663 665 679 681 732 749
                                           769 812 827 829 853 861 869 877 885 924
EEPROM_RDY . . . C ADDR   02A5H   R      SEG=SERIAL_BOOT_CODE  647#
EEPROM_SELECT. .     NUMB   0040H   A        68# 644 658 946 1212
ENTRY. . . . . . C ADDR   0023H   R      SEG=PROFILE_FLT_CODE  920 965# 1249
EOT_CMD. . . . . C ADDR   0216H   R      SEG=SERIAL_BOOT_CODE  486 565#
EOT_CODE . . . .     NUMB   00A8H   A        106# 483 648 664 680 731 748 788 811 828 852 860 868 876 884 923
ERR_CODE . . . .     NUMB   00FFH   A        105# 553
ERR. . . . . . . . C ADDR   01F5H   R      SEG=SERIAL_BOOT_CODE  408 412 414 417 421 423 426 428 546# 569 616
ES . . . . . . . . B ADDR   00A8H.4 A        245 247 255
ESC_CMD. . . . . C ADDR   01AAH   R      SEG=SERIAL_BOOT_CODE  481 502# 550
ESC_CODE . . . .     NUMB   007FH   A        104# 480 503 549
EX0. . . . . . . . B ADDR   00A8H.0 A        172 253
EXTI0. . . . . . C ADDR   0003H   A        143
EXTI1. . . . . . C ADDR   0013H   A        146
FIFO14 . . . . . C ADDR   016DH   R      SEG=SERIAL_BOOT_CODE  459 461# 466
FLIGHT_BOOT. . . C ADDR   0026H   R      SEG=SERIAL_BOOT_CODE  216 246#
FLIGHT_PROFILE . C ADDR   0005H   R      SEG=PROFILE_FLT_CODE  911 945#
FLT_DELAY. . . . C ADDR   003FH   R      SEG=SERIAL_BOOT_CODE  255 265#
FORCE_RETURN . . C ADDR   020EH   R      SEG=SERIAL_BOOT_CODE  558#
FRAME_WORD_0 . . X ADDR   0FA0H   A        111#
FRAME_WORD_1 . . X ADDR   0FA1H   A        112#
G. . . . . . . . .   NUMB   0047H   A        87# 411
GET_MON. . . . . C ADDR   02E6H   R      SEG=SERIAL_BOOT_CODE  677 685#
GET_NEXT17 . . . C ADDR   002DH   R      SEG=PROFILE_FLT_CODE  972# 977
```

```
N A M E          T Y P E  V A L U E      ATTRIBUTES AND REFERENCES

GT_0 . . . . . . . C ADDR   0138H   R      SEG=SERIAL_BOOT_CODE  425 427#
GT1. . . . . . . . C ADDR   01E4H   R      SEG=SERIAL_BOOT_CODE  531 532 535#
HV_1_MON . . . .     NUMB   0002H   A        91# 1105
HV_2_MON . . . .     NUMB   0003H   A        92# 1114
IE0. . . . . . . . B ADDR   0088H.1 A        174
IF_FLT . . . . . . C ADDR   0076H   A        196# 199 205
ION_CONT . . . . . B ADDR   0090H.7 A        54# 875 883 1019
IS_FLT . . . . . . B ADDR   0022H.3 A        45#
IS_LT_9. . . . . . C ADDR   0132H   R      SEG=SERIAL_BOOT_CODE  422 424#
IS_LT_6. . . . . . C ADDR   011FH   R      SEG=SERIAL_BOOT_CODE  413 415#
IT0. . . . . . . . B ADDR   0088H.0 A        171 251
JMP_TO_PROFILE . . C ADDR   049CH   R      SEG=SERIAL_BOOT_CODE  931 935#
KEEP_SAFETY. . . . C ADDR   002CH   R      SEG=SERIAL_BOOT_CODE  247 249#
LSB_DC . . . . .     NUMB   0060H   A        73# 745 1013 1195
LSB_RF . . . . .     NUMB   0058H   A        71# 728 1007 1189
LT_9 . . . . . . . C ADDR   012DH   R      SEG=SERIAL_BOOT_CODE  420 422#
LT_6 . . . . . . . C ADDR   011AH   R      SEG=SERIAL_BOOT_CODE  411 413#
MAY_FLIGHT . . . . C ADDR   0087H   A        213#
MON_ALE. . . . . . B ADDR   0090H.4 A        50# 168 689 690 1254 1258 1259
MON_OE . . . . . . B ADDR   0090H.2 A        52# 167 698 700 1271 1273 1286
MON_SELECT . . .     NUMB   0000H   A        81# 686 696 1255
MON_STRT . . . . . B ADDR   0090H.3 A        51# 691 692 1261 1262
MSB_DC . . . . .     NUMB   00E0H   A        74# 741 1010 1192
MSB_RF . . . . .     NUMB   00D8H   A        72# 724 1004 1186
MULTI_AMU. . . . . B ADDR   0022H.5 A        47# 906 918 947 966 1179 1182 1213
NEXT_STEP. . . . . C ADDR   0477H   R      SEG=SERIAL_BOOT_CODE  912# 921
NIN. . . . . . .     NUMB   0039H   A        89# 420
NO_ERR . . . . . . C ADDR   0110H   R      SEG=SERIAL_BOOT_CODE  406 409#
NO_ESC_ERR . . . . C ADDR   01FEH   R      SEG=SERIAL_BOOT_CODE  549 551#
NO_OF_CMDS . . .     NUMB   000EH   A        571#
NOT_A_F. . . . . . C ADDR   0318H   R      SEG=SERIAL_BOOT_CODE  708 712#
NOT_EEPROM_RESET   C ADDR   01AAH   R      SEG=PROFILE_FLT_CODE  1209 1211#
NOT_END_PROFILE.   C ADDR   016FH   R      SEG=PROFILE_FLT_CODE  1177 1181#
NOT_EOT. . . . . . C ADDR   0199H   R      SEG=SERIAL_BOOT_CODE  483 492#
NOT_ESC. . . . . . C ADDR   0186H   R      SEG=SERIAL_BOOT_CODE  480 482#
NOT_LT_A . . . . . C ADDR   0125H   R      SEG=SERIAL_BOOT_CODE  416 418#
NOT_MULTI_AMU. . . C ADDR   0198H   R      SEG=PROFILE_FLT_CODE  1176 1180 1199#
NOT_OVERFLOW . . . C ADDR   022CH   R      SEG=PROFILE_FLT_CODE  1292 1294#
OD . . . . . . . . B ADDR   0090H.6 A        60# 164 326
P0 . . . . . . . . D ADDR   0080H   A        185 323 354 687 697 699 721 722 726 730 738 739 743 747 782 783 787 805
                                             806 810 951 981 986 991 996 1003 1006 1009 1012 1020 1023 1026 1029 1185
                                             1188 1191 1194 1239 1241 1243 1245 1268 1270 1272 1274 1289 1296 1300
P1 . . . . . . . . D ADDR   0090H   A        50 51 52 53 54 57 59 60 1285 1303
P2 . . . . . . . . D ADDR   00A0H   A        58 163 186 688 693 696 701 723 724 725 727 728 729 740 741 742 744 745
                                             746 784 785 786 807 808 809 949 950 982 983 987 988 992 993 997 998 1004
```

```
NAME             TYPE   VALUE       ATTRIBUTES AND REFERENCES

                                       1005 1007 1008 1010 1011 1013 1014 1021 1022 1024 1025 1027 1028 1030
                                       1186 1187 1189 1190 1192 1193 1195 1196 1240 1242 1244 1246 1256 1257
                                       1260 1267 1269 1275 1288 1295 1297 1298 1301
P3 . . . . . . . D ADDR   00B0H   A      55 56
PCM_LOAD . . . .   NUMB   0078H   A      77# 1297
PCM_POINT. . . .   NUMB   0008H   A      17# 183 252 960 1235
PCM_ROUTINE. . . C ADDR   0215H   R    SEG=PROFILE_FLT_CODE  144 1280#
PRE_ESC. . . . . C ADDR   01C1H   R    SEG=SERIAL_BOOT_CODE  509 513#
PROFILE_FLT_CODE C SEG    0244H        REL=UNIT  123# 938
PROFILE_RETURN . C ADDR   01ECH   R    SEG=PROFILE_FLT_CODE  1248 1250#
PROFILE. . . . . X ADDR   0FB0H   A      118#
PS . . . . . . . B ADDR   00B8H.4 A      243
PSW. . . . . . . D ADDR   00D0H   A      387 394 437 489 498 560 1281 1304
PX0. . . . . . . B ADDR   00B8H.0 A      170 250
Q_BIAS . . . . . D ADDR   000DH   A      24# 1017 1020 1173
Q_FULL . . . . . C ADDR   01E9H   R    SEG=SERIAL_BOOT_CODE  524 540# 540
Q_NOT_EMPTY. . . C ADDR   015AH   R    SEG=SERIAL_BOOT_CODE  446 449#
Q_PTR. . . . . . D ADDR   004FH   A      36# 235 445 458 469 522 523 529 530 913
QUEUE. . . . . . D ADDR   0039H   A      35# 179 182 183 252 268 271 450 455 456 523 526 960 1044 1055 1070 1071
                                       1077 1078 1104 1113 1122 1131 1140 1149 1158 1167 1173 1228 1232 1235
                                       1241 1245 1247 1292 1293
R12_BIT_DECODE . C ADDR   036AH   R    SEG=SERIAL_BOOT_CODE  720 737 755#
RAT_H. . . . . . D ADDR   000BH   A      22# 971 1009 1077 1191
RAT_L. . . . . . D ADDR   000CH   A      23# 1012 1078 1194
RCV_ERR. . . . . C ADDR   010DH   R    SEG=SERIAL_BOOT_CODE  403 407#
RECIEVE. . . . . C ADDR   00FDH   R    SEG=SERIAL_BOOT_CODE  391 399#
RESET. . . . . . C ADDR   0000H   A      218
RF_HV. . . . . . B ADDR   00B0H.5 A      55# 166 381 859 867
RF_MON . . . . .   NUMB   0006H   A      90# 1096
RI . . . . . . . B ADDR   0098H.0 A      242 391 400
ROOM_IN_Q. . . . C ADDR   01D3H   R    SEG=SERIAL_BOOT_CODE  523 525#
RS0. . . . . . . B ADDR   00D0H.3 A      161 389 1284
RS1. . . . . . . B ADDR   00D0H.4 A      162 390 1283
SAVE_BIT . . . . B ADDR   0022H.7 A      49#
SBUF . . . . . . D ADDR   0099H   A      401 450
SCON . . . . . . D ADDR   0098H   A      239
SELECT . . . . . B ADDR   00A0H.6 A      58#
SERIAL_BOOT_CODE C SEG    049FH        REL=UNIT  122# 228
SERIAL_BOOT. . . C ADDR   0000H   R    SEG=SERIAL_BOOT_CODE  189 212 230#
SERIAL_STEP. . . B ADDR   0022H.4 A      46# 901 925 1248
SERIAL . . . . . C ADDR   00E8H   R    SEG=SERIAL_BOOT_CODE  152 786#
SET_NEW_VALS . . C ADDR   0029H   R    SEG=PROFILE_FLT_CODE  966 968#
SINT . . . . . . C ADDR   0023H   A      151
SP . . . . . . . D ADDR   0081H   A      187 505 559 710
STACK. . . . . . D ADDR   0050H   A      37# 187 505 559
```

```
N A M E             T Y P E   V A L U E       ATTRIBUTES AND REFERENCES

STEP_EXIT. . . . .  C ADDR    049BH   R       SEG=SERIAL_BOOT_CODE  819 822#
STROBE . . . . . .  B ADDR    0090H.5 A         59# 165 724 725 755 756
TEMP_MON . . . . .    NUMB    0007H   A         97# 1159
TEST_TIME. . . . .    NUMB    0005H   A         192# 193 194 195
TH1. . . . . . . .  D ADDR    008DH   A         228
TI . . . . . . . .  B ADDR    0098H.1 A         241 392 534 541 814
TIME_BIT . . . . .  B ADDR    0020H.6 A         48# 284 286 287 308 311 311 339 341 342 369 371 372 847 851 1035 1036
                                                1050 1056 1061 1066 1077 1081 1092 1099 1108 1117 1124 1135 1144 1153
                                                1162 1169 1204 1207 1226 1229
TL1. . . . . . . .  D ADDR    008BH   A         227
TMOD . . . . . . .  D ADDR    0089H   A         226
TO_BOOT_IT . . . .  C ADDR    00B4H   A         200 208#
TO_BOOT. . . . . .  C ADDR    0075H   A         140 158#
TO_DO_IT04 . . . .  C ADDR    0050H   R       SEG=SERIAL_BOOT_CODE  280# 291
TO_DO_IT05 . . . .  C ADDR    0070H   R       SEG=SERIAL_BOOT_CODE  304# 314
TO_DO_IT0A . . . .  C ADDR    00AAH   R       SEG=SERIAL_BOOT_CODE  335# 345
TO_DO_IT0F . . . .  C ADDR    00CEH   R       SEG=SERIAL_BOOT_CODE  365# 375
TO_DONE04. . . . .  C ADDR    0064H   R       SEG=SERIAL_BOOT_CODE  282 291#
TO_DONE09. . . . .  C ADDR    0085H   R       SEG=SERIAL_BOOT_CODE  306 315#
TO_DONE0E. . . . .  C ADDR    00B2H   R       SEG=SERIAL_BOOT_CODE  337 346#
TO_DONE13. . . . .  C ADDR    00DDH   R       SEG=SERIAL_BOOT_CODE  367 376#
TO_DONE16. . . . .  C ADDR    0173H   R       SEG=SERIAL_BOOT_CODE  461 467#
TO_ERR . . . . . .  C ADDR    0117H   R       SEG=SERIAL_BOOT_CODE  409 412#
TO_RESET_PAYLOAD    C ADDR    008DH   A         215 217#
TOP_OF_IMP_TBLE.    C ADDR    0227H   R       SEG=SERIAL_BOOT_CODE  575 577#
TR1. . . . . . . .  B ADDR    0088H.6 A         240
TRASH_EE . . . . .  C ADDR    0FA5H   A         117#
TST. . . . . . . .  B ADDR    00B0H.4 A         56# 196 197
VALID. . . . . . .  C ADDR    013DH   R       SEG=SERIAL_BOOT_CODE  419 427 429#
VCC_MON. . . . . .    NUMB    0001H   A         95# 1141
WAIT_ONE_MS02. . .  C ADDR    0059H   R       SEG=SERIAL_BOOT_CODE  285# 288
WAIT_ONE_MS07. . .  C ADDR    007AH   R       SEG=SERIAL_BOOT_CODE  309# 312
WAIT_ONE_MS0C. . .  C ADDR    00A7H   R       SEG=SERIAL_BOOT_CODE  340# 343
WAIT_ONE_MS11. . .  C ADDR    00D2H   R       SEG=SERIAL_BOOT_CODE  370# 373
WAIT_T_1 . . . . .  C ADDR    00AAH   R       SEG=PROFILE_FLT_CODE  967 1032#
WAIT_T_2 . . . . .  C ADDR    0025H   R       SEG=PROFILE_FLT_CODE  1049#
WAIT . . . . . . .  C ADDR    003AH   R       SEG=SERIAL_BOOT_CODE  256# 262
WORD_IT_WAIT . . .  C ADDR    01B7H   R       SEG=PROFILE_FLT_CODE  1198 1220#
XER. . . . . . . .  B ADDR    0020H.1 A         47# 232 447 472 514 522 810 817 819
XFR. . . . . . . .  B ADDR    0090H.0 A         57# 169 732 734 751 751 848 1032 1033 1034
XMIT . . . . . . .  C ADDR    014FH   R       SEG=SERIAL_BOOT_CODE  392 442# 541
ZER. . . . . . . .    NUMB    0030H   A         59# 425 715


REGISTER BANK(S) USED: 0 1
```

```
MCS-51 MACRO ASSEMBLER      (LIFO  SUPER ARCAS)                                              16 OCT 85  PAGE   75

N A M E              T Y P E   V A L U E        ATTRIBUTES AND REFERENCES

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

IX. PERSONNEL

A list of the engineers who contributed to the work reported is given below:

J. Spencer Rochefort, Professor of Electrical and Computer Engineering and Principal Investigator.

Raimundas Sukys, Senior Research Associate, Engineer.

X. RELATED CONTRACTS AND PUBLICATIONS

F19628-74-C-0042	1 Sept. 1973 through Oct. 1976
F19628-76-C-0256	1 Aug. 1976 through 31 Oct. 1978
F19628-78-C-0218	15 Sept. 1978 through Sept. 1981
F19628-81-C-0162	15 Sept. 1981 through Sept. 1985

Raimundas Sukys, Steven Goldberg, "Control Circuits for Rocket Payload Neutralization Experiment and Other Topics", Scientific Report No. 1 for Contract F19628-74-C-0042, October 1974, AFGRL-TR-74-0580, ADA008039.

R. Sukys, J. Spencer Rochefort, S. Goldberg, "Bias and Signal Processing Circuits for a Mass Spectrometer in the Project EXCEDE: SWIR Experiment", Scientific Report No. 2 for Contract F19628-74-C-0042, October 1975, AFGL-TR-76-0060, ADA026514.

J. Spencer Rochefort, Raimundas Sukys, "Instrumentation Systems for Maas Spectrometers", Final Report for Contract F19628-74-C-0042, September 1976, AFGL-TR-76-0200, ADA032313.

J. Spencer Rochefort, Raimundas Sukys, "A Digital Control Unit for a Rocket Borne Quadrupole Mass Spectrometer", Scientific Report No. 1, for Contract F19628-76-C-0256, April 1978, AFGRL-TR-78-0106, ADA57251.

J. Spencer Rochefort, Raimundas Sukys, "Electronics for a Rocket Borne Quadrupole Cluster Ion Mass Filter", Final Report for Contract F19628-76-C-0256, October 1978, AFGL-TR-78-0292, ADA066289.

Gerousis, V.S., "A Programmable Control Unit for a Balloon-Borne Mass Spectrometer Based on Intel 8085A Microprocessor", Scientific Report No. 1, Contract AF19628-78-C-0218, September 1979, AFGL-TR-79-0225, ADA106398.

Palasek. T., "An RF Oscillator for Rocket-Borne and Balloon-Borne Mass Spectrometers", Scientific Report No. 2, Contract AF19628-78-C-0218, September 1979, AFGL-TR-79-0226, ADA078797.

Sukys, R. and Rochefort, J.S. "Control and Data Transmission System for a Balloon-Borne Ion Mass Spectrometer", Proceedings International Telemetering Conference, October 1980, Vol. XVI, pp. 335-341.

Sukys, R. and Rochefort, J.S. "Control and Data Transmission System for a Balloon-Borne Ion Mass Spectrometer", Scientific Report No. 3 Contract AF19628-78-C-0218, October 1980, AFGL-TR-81-0202, ADA104512.

J. Spencer Rochefort, Raimundas Sukys, "Control Electronics for Airborne Quadrupole Ion Mass Spectrometer", Final Report for Contract F19628-78-C-0218, October 1981, AFGL-TR-82-0056, ADA115399.

Sukys, R. and Rochefort, J.S. "GSE for Balloon Borne I.M.S.: Decommutator and D/A Units", Scientific Report No.1 for Contract F19628-81-C-0162, October 1982, AFGL-TR-83-0095, ADA131845.

Sukys, R. and Rochefort, J.S. "Downrigger Instrumentation to Record Thermosonde Data", Scientific Report No. 2 for Contract F19628-81-C-0162, October 1983, AFGL-TR-85-0085, ADA161748.

Sukys, Raimundas, "Control and Signal Conditioning Circuits for E.I.R.M.A., Scientific Report No. 3 for Contract F19628-81-C-0162, October 1984, AFGL-TR-85-0117, ADA166683.

ATE
MED
-8